www.freeservicemanuals.info 11/14/2023

PHILIPS TECHNISCH TIJDSCHRIFT

JAARGANG 29, 1968, no. 11

Een experimentele kleine kleurentelevisiecamera

P. H. Broerse, J. H. T. van Roosmalen en S. L. Tan

In het onderstaande artikel vindt men een beschrijving van de draagbare kleurentelevisiecamera waarvan reeds enige maanden geleden in dit tijdschrift een afbeelding werd getoond. De constructie van een dergelijke kleine camera is mogelijk geworden door de vervaardiging van een (experimentele) „Plumbicon" opneembuis die kleiner is dan de „Plumbicon" buizen die in de studiocamera voor kleurentelevisie worden gebruikt. Gebleken is dat de kwaliteit van de beelden, opgenomen met deze miniatuurcamera, een vergelijking met die van de thans in gebruik zijnde studiocamera goed kan doorstaan.

Het toepassingsgebied van televisiecamera's is al sinds jaren niet meer beperkt tot het gebruik in studio's en bij reportages. Reeds vroegtijdig heeft men het nut ingezien van televisie in de industrie, bij het onderwijs en voor vele andere doeleinden. Een frappant voorbeeld uit het medisch onderwijs is het zichtbaar maken van medische ingrepen voor een groot aantal studenten [1]. Van meer recente datum is het gebruik van een televisiecamera in combinatie met een endoscoop, waardoor artsen de mogelijkheid krijgen om inwendige organen van patiënten op een televisiescherm zichtbaar te maken. Ook op andere gebieden van het onderwijs, bij wetenschappelijk onderzoek en in de industrie dringt het gebruik van televisie steeds verder door. Als voorbeeld noemen wij de mogelijkheid om een camera te koppelen met een microscoop, een combinatie die in de elektronische industrie een waardevol hulpmiddel kan vormen bij het controleren van de fabricage van geïntegreerde schakelingen.

In enkele van de hierboven genoemde gevallen is de informatie die wordt verkregen met behulp van zwart-wit-televisie voldoende. Vaak echter is de weergave van kleuren van essentieel belang; met name is dit het geval bij medische toepassingen [2]. Speciaal hier ontstaan echter door de grote afmetingen en het gewicht van een kleurencamera moeilijkheden. Om geen hinder te veroorzaken bij een operatie heeft men een kleine lichte camera nodig. Ook in gevallen waar men camera's wil koppelen aan andere instrumenten bestaat deze wens, en bij de televisie-omroep kan een kleine draagbare camera voor kleurentelevisie nieuwe mogelijkheden scheppen.

Met het oog op deze behoefte is een experimentele kleurencamera gemaakt waarbij men getracht heeft met de thans ter beschikking staande middelen de afmetingen en het gewicht zo veel mogelijk te beperken.

Deze camera, die 3 kg weegt en die, wat grootte betreft, kan worden vergeleken met een 16-mm-filmcamera, is enkele maanden geleden aan vakkringen gedemonstreerd [3]. In dit artikel zullen wij enkele problemen bespreken die zich voordeden bij de vervaardiging van de hiervoor bestemde opneembuizen en bij de constructie van de camera. *Fig. 1* toont de camera te zamen met een microscoop in gebruik voor het controleren van een geïntegreerde schakeling.

De opneembuis

Voorop stond dat de camera zoveel mogelijk dezelfde gunstige eigenschappen moest hebben als de grote camera met „Plumbicon" opneembuizen die eerder in dit tijdschrift is beschreven [4]. Nu worden de afmetingen van een kleurentelevisiecamera met drie opneembuizen voor een belangrijk deel bepaald door de grootte van deze buizen en van de deflexiespoelen. De minia-

Drs. P. H. Broerse, J. H. T. van Roosmalen en ir. S. L. Tan, medewerkers van Philips' Natuurkundig Laboratorium, Eindhoven.

[1] Zie b.v. Philips techn. T. **11**, 90, 1949.
[2] W. Holm en F. H. J. van der Poel, Kleurentelevisie in dienst van het medisch onderwijs, Philips techn. T. **20**, 343-346, 1958.
[3] Philips techn. T. **29**, 191, 1968 (no. 6).
[4] Zie H. Breimer, W. Holm en S. L. Tan, Een kleurentelevisiecamera met „Plumbicon" opneembuizen, Philips techn. T. **28**, 346-361, 1967 (no. 11).

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

tuurcamera kon dan ook pas worden geconstrueerd na de vervaardiging van een experimentele „Plumbicon" buis van veel kleinere afmetingen dan de tot dusver gebruikte. De standaard „Plumbicon" buis heeft een diameter van 30 mm en een lengte van 200 mm; bij de miniatuuruitvoering zijn deze afmetingen resp. 16 en 130 mm. Het beeldformaat van 8,4×6,3 mm (beeld-

De constructie van de miniatuur „Plumbicon" buis is in principe gelijk aan die van de grotere, in het aangehaalde artikel [5] beschreven uitvoering. De voornaamste problemen die optraden bij de verkleining van de afmetingen hadden betrekking op de handhaving van een voldoend groot scheidend vermogen en op het superponeren van de beelden van de drie in de camera

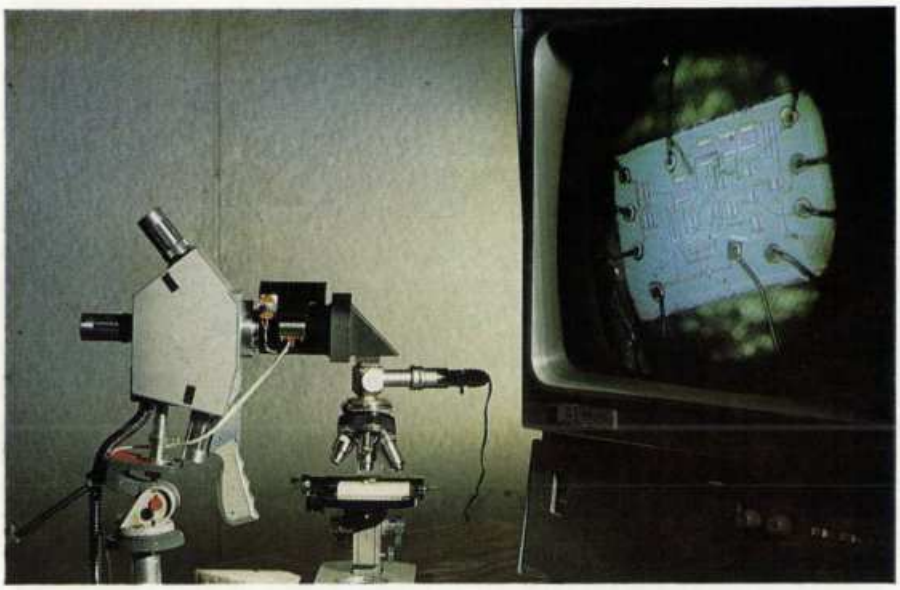


Fig. 1. De experimentele kleine camera, direct gekoppeld met een microscoop. Op de monitor is een geïntegreerde schakeling zichtbaar die met de microscoop wordt gecontroleerd.

diagonaal 10,5 mm) is lineair een factor twee kleiner dan bij de standaardbuis. *Fig. 2a,b* toont de beide „Plumbicon" buizen, resp. zonder en met hun deflexiespoelen.

De focussering van de elektronenbundel geschiedt elektrostatisch. De voordelen van elektrostatische ten opzichte van magnetische focussering zijn reeds in een vroeger artikel uiteengezet, aan de hand van een buis met standaardafmetingen [5]. Bij het streven naar ruimte- en gewichtsbesparing weegt uiteraard het overbodig zijn van lijvige focusseerspoelen zwaar; ook andere voordelen van elektrostatische focussering spelen echter een rol. Wij noemen het feit dat bij de focussering geen draaiing, vergroting of verkleining van het beeld optreedt, en verder de onderlinge onafhankelijkheid van deflexie en focussering, waardoor het beeld gemakkelijk over het gehele oppervlak scherp te stellen is. Verder is van belang dat voor elektrostatische focussering geen vermogen nodig is.

geplaatste buizen. In de volgende paragrafen zullen wij hierop nader ingaan.

Fig. 3 is een schematische doorsnede van de buis. De elektronenbundel wordt geleverd door het *triodekanon Tr*, dat bestaat uit de geïmpregneerde katode *K* (vermogen 2 W), de stuurelektrode („rooster") *G* en de anode A_1. De bundel wordt samengebracht op de fotogeleidende laag *T* door de focusseerlens, die wordt

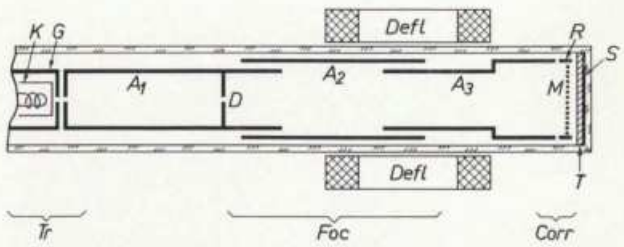


Fig. 3. Schematische bouw van een „Plumbicon" buis met elektrostatische focussering. *Tr* triodekanon, bestaande uit de katode *K*, de stuurelektrode *G* en de anode A_1. De focusseerlens *Foc* wordt gevormd door de elektrodes A_1, A_2 en A_3. *D* diafragma. De ring *R* en het gaas *M* vormen te zamen met A_3 de correctielens *Corr*. *T* fotogeleidende laag. *S* signaalplaat. *Defl* deflexiespoelen.

[5] J. H. T. van Roosmalen, Experimentele elektrostatisch gefocusseerde „Plumbicon" buizen, Philips techn. T. **28**, 64-70, 1967 (no. 2).

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

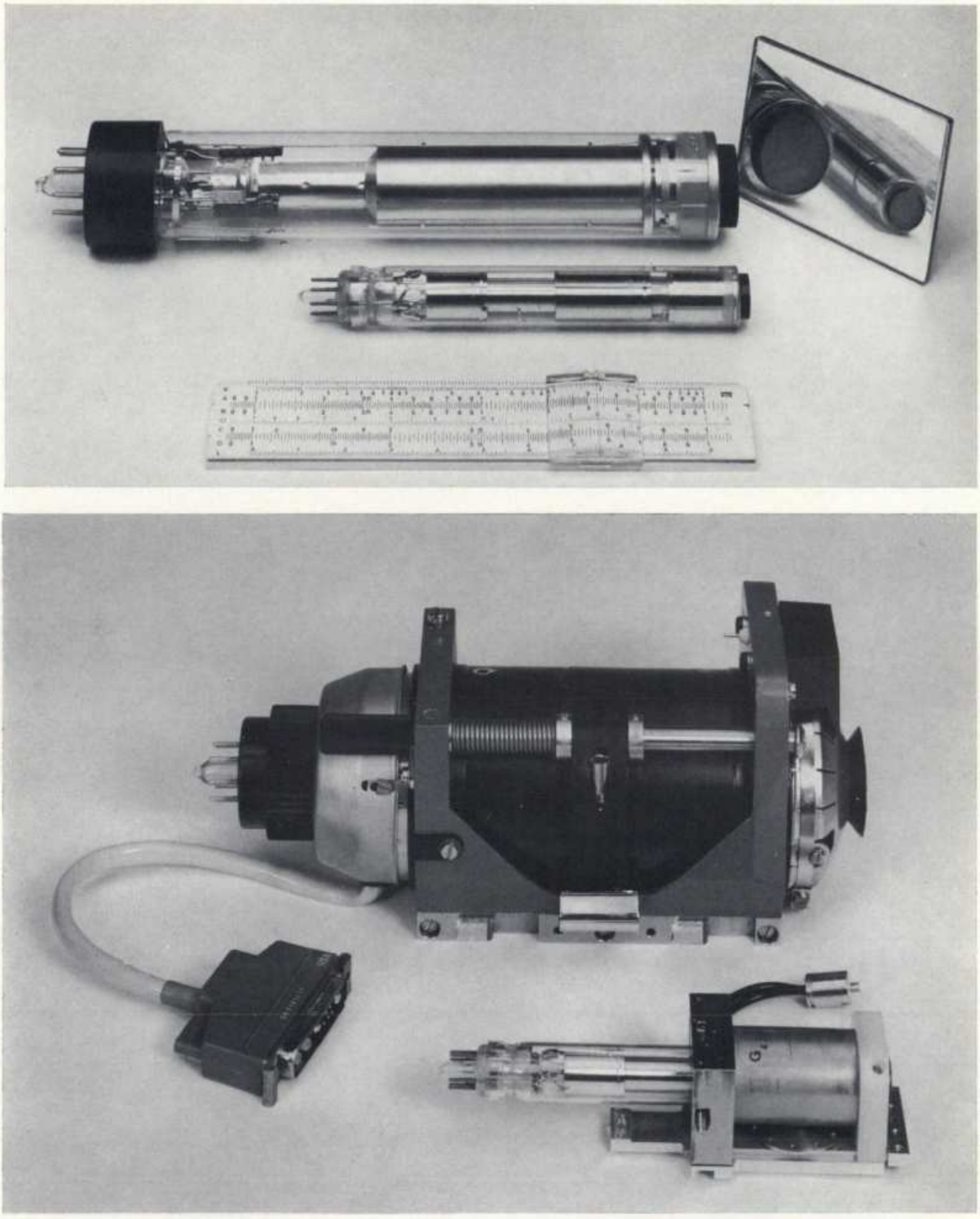


Fig. 2. *a*) Standaard en miniatuur „Plumbicon" buis. De afmetingen van de buizen zijn resp. 200 × 30 mm en 130 × 16 mm. *b*) De beide buizen geplaatst in hun deflexiespoelen.

gevormd door de cilinders A_1, A_2 en A_3. In A_1 bevindt zich een diafragma D. Voorts is achter de focusseerlens een *correctielens* aangebracht, die bestaat uit het rechter deel van A_3, de ring R en het gaas M. De potentiaal van R en M is hoger dan die van A_3. Deze lens buigt de elektronenbundel zodanig af dat de elektronen de fotogeleidende laag vrijwel loodrecht treffen. Voor een beschrijving van de stooreffecten die optreden als dit laatste niet het geval is, verwijzen wij weer naar het aangehaalde artikel [5].

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

De focussering

Door de lenswerking van K, G en A_1 worden de geëmitteerde elektronen gefocusseerd in de zg. bundelknoop („cross-over"). Deze wordt door de focusseerlens afgebeeld op de fotogeleidende laag. Om een goed scheidend vermogen te krijgen moet deze afbeelding klein zijn, en wel des te kleiner naarmate het beeldformaat kleiner is.

Evenals bij de ontwikkeling van de grote „Plumbicon" buis met elektrostatische focussering zijn de afmetingen, de onderlinge plaats en de potentialen van de elektrodes vastgesteld na het berekenen van elektronenbanen bij verschillende configuraties. In *fig. 4* zijn voor de uiteindelijke uitvoering een aantal equipotentiaalvlakken weergegeven, alsmede enkele elektronenbanen van de niet-gedeflecteerde bundel.

hierbij het voor de deflexie nodige vermogen toeneemt en het gevaar voor overslag tussen R en A_3 groter wordt. De bundelstroom I_b moet blijkens de formule voor een kleine waarde van d_M klein zijn, doch voor het snel ontladen van het scherm (dus een geringe traagheid) is een niet te kleine waarde gewenst. Verder zijn de maximaal toelaatbare stroomdichtheid j_0 en de temperatuur T_k door het toegepaste type katode vastgelegd. Maken we ten slotte α_M groot, dan ontstaan moeilijkheden door sferische aberratie van de focusseerlens en ook door de optredende defocusserende werking van de deflexiespoelen.

Door de *sferische aberratie* van de focusseerlens worden de elektronen verspreid over een bepaalde verstrooiingscirkel. De diameter Δd_M hiervan ter plaatse van het gaas is evenredig met de derde macht van de dia-

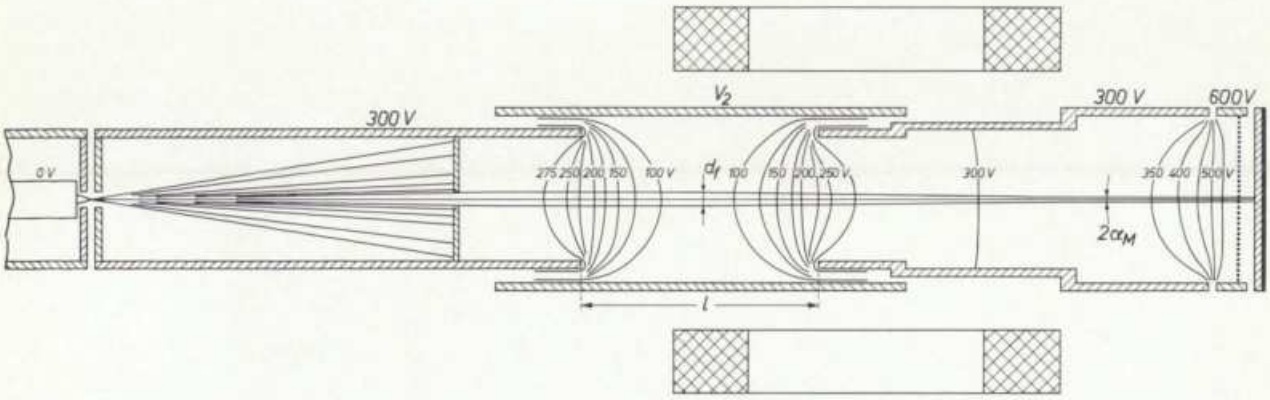


Fig. 4. Enkele equipotentiaalvlakken en elektronenbanen bij een niet-gedeflecteerde bundel. Bij de elektrodes A_1, A_3 en R (zie fig. 3) zijn de spanningen ten opzichte van de katode aangegeven. Met de spanning V_2 van A_2 wordt de bundel gefocusseerd. d_f bundeldiameter in het midden van de focusseerlens. $2\alpha_M$ openingshoek van de bundel ter plaatse van het gaas. l afstand tussen A_1 en A_3.

Voor het verkrijgen van een kleine afbeelding van de bundelknoop werken, wat het elektronenoptische gedeelte betreft, de thermische spreiding van de elektronen, de sferische aberratie van de focusseerlens en de onderlinge afstoting van de elektronen ongunstig.

Ten gevolge van de *thermische spreiding* heeft de bundel ter plaatse van het gaas een diameter

$$d_M = \frac{2}{\alpha_M} \sqrt{\frac{I_b\, k\, T_k}{\pi\, j_0\, e\, V_M}}.$$

Hierin is α_M de halve openingshoek van de bundel ter plaatse van het gaas, V_M de potentiaal van het gaas, I_b de bundelstroom na het diafragma en j_0 de stroomdichtheid in het midden van de katode. Verder is k de constante van Boltzmann, T_k de katodetemperatuur en e de lading van het elektron. De in deze formule voorkomende grootheden willen we zo kiezen dat d_M zo klein mogelijk wordt, doch hierbij zijn wij voor ieder van deze grootheden aan een grens gebonden. De spanning V_M kan niet onbeperkt worden opgevoerd, omdat

meter d_f van de bundel in het midden van de focusseerlens:

$$\Delta d_M = B d_f^3.$$

De aberratiecoëfficiënt B willen we uiteraard zo klein mogelijk maken. Een in dit opzicht gunstige configuratie werd weer gevonden door het berekenen van de elektronenbanen. De resultaten van een aantal van deze berekeningen zijn samengevat in *fig. 5*, waar B is uitgezet als functie van de afstand l tussen A_1 en A_3. We zien dat B daalt met toenemende waarde van l. Omdat een vergroting van l een langere buis nodig maakt, hebben wij ons in dit opzicht moeten beperken en is de grootte van l op 16 mm vastgesteld.

Aan de verschillende tegengestelde eisen kan beter worden voldaan als men in de anode A_1 een diafragma plaatst (zie fig. 3 en fig. 4). Hierdoor is nl. een gunstiger verhouding van I_b en j_0 mogelijk. Op de vaststelling van de plaats en de grootte van dit diafragma komen wij nog terug.

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

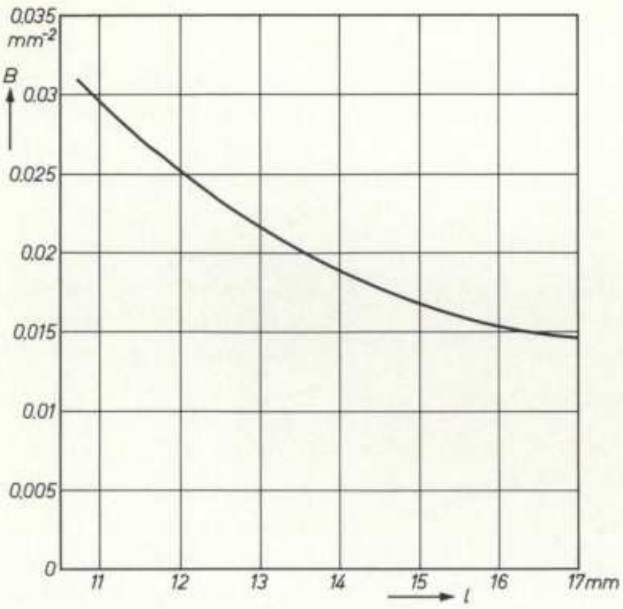


Fig. 5. De sferische-aberratiecoëfficiënt B als functie van de afstand l tussen de anodes A_1 en A_3.

De correctielens

De potentiaal van het gaas M en de ring R is tweemaal zo hoog als die van de cilinders A_1 en A_3 (600 resp. 300 V). Wanneer nu de deflexiespoelen zodanig worden geplaatst dat het deflexiepunt (het punt op de as waar de gedeflecteerde bundel vandaan schijnt te komen) op een afstand van het gaas ligt die ongeveer gelijk is aan 2× de diameter van het gaas, dan wordt de gedeflecteerde bundel door de correctielens zodanig afgebogen dat zijn richting weer evenwijdig is met de as van de buis [6]. De hierbij optredende tonvormige vertekening wordt gecompenseerd door een bepaalde vormgeving van de deflexiespoelen. *Fig. 6* toont het verloop van de as van de bundel in een aantal gedeflecteerde standen.

In verband met de kleinere afbeelding op de fotogeleidende laag was het nodig het gaasje, dat in de standaardbuis een spoed van 36 μm heeft, te vervangen door een gaasje met een spoed van slechts 18 μm. De spijldikte is 5 à 6 μm. Dit fijnere gaas laat, nadat het aan een etsprocédé is onderworpen, ongeveer 45% van de elektronenbundel door.

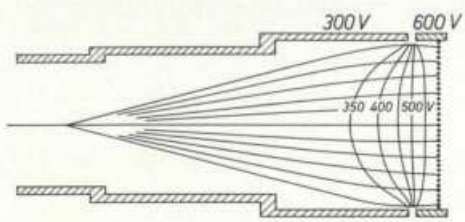


Fig. 6. Het verloop van de as van de bundel in enkele gedeflecteerde standen. De correctielens doet de elektronen loodrecht op de fotogeleidende laag landen.

De openingshoek van de bundel

Een belangrijk punt bij het ontwerp was de vaststelling van de plaats en de grootte van het diafragma. Hoe verder dit zich, bij een constante diameter, bevindt van de bundelknoop, des te kleiner is de openingshoek en des te „slanker" is de bundel. Een slanke bundel heeft het voordeel dat de sferische aberratie van de lens en de defocusserende werking van de deflexiespoelen kleiner worden. Anderzijds neemt dan echter bij een constante stroom de invloed van de onderlinge afstoting van de elektronen toe. Men kan dus verwachten dat met betrekking tot het scheidend vermogen een optimale waarde van de openingshoek van de bundel bestaat. Inderdaad is dit uit experimenten gebleken. Hiervoor is een aantal buizen gemaakt waarin het diafragma verschuifbaar was tussen afstanden van 6 en 21 mm van de bundelknoop. (De diameter van de opening was 0,35 mm.) Het was hierdoor mogelijk de bundeldiameter in het midden van de focusseerlens te veranderen van 2,4 tot 0,7 mm. Een aantal resultaten van met deze buizen uitgevoerde metingen zijn verzameld in *fig. 7*.

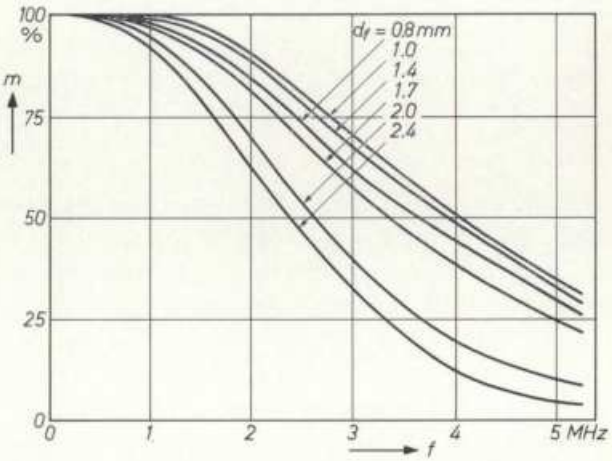


Fig. 7. De modulatiediepte m als functie van de frequentie f bij enkele waarden van de diameter d_f van de bundel in het midden van de focusseerlens. Een optimale waarde van m wordt ongeveer bereikt bij $d_f = 1{,}0$ mm.

Uitgezet is hier de modulatiediepte m [7] als functie van de frequentie, met als parameter de diameter d_f van de bundel in het midden van de lens. We zien dat er inderdaad een waarde van d_f is waarbij m het grootst is ($d_f \approx 1{,}0$ mm). Bij een frequentie van 5 MHz is dan de modulatiediepte 35%.

Fig. 7 geeft de modulatiediepte weer in het midden van het scherm en men kan zich natuurlijk afvragen hoe het scheidend vermogen op andere plaatsen is. Een

[6] J. H. T. van Roosmalen, Ned. octrooiaanvrage no. 6513751.

[7] Voor de wijze waarop men het scheidend vermogen hierin uitdrukt, zie: E. F. de Haan, A. van der Drift en P. P. M. Schampers, Het „Plumbicon", een nieuwe televisie-opneembuis, Philips techn. T. **25**, 277-295, 1963, speciaal fig. 17.

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

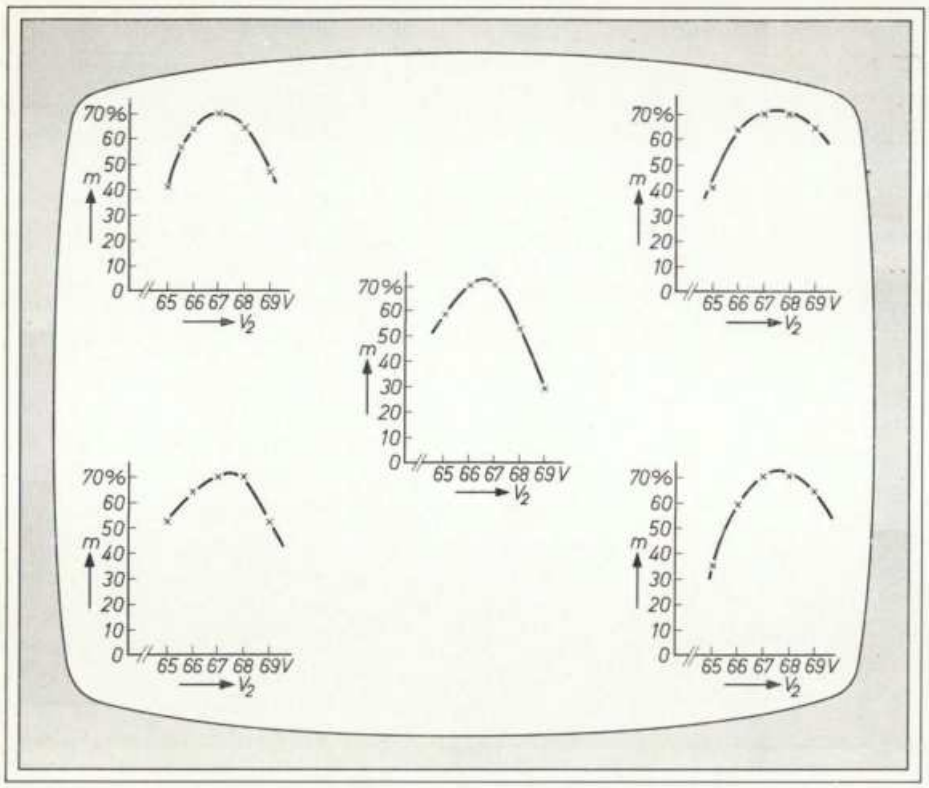


Fig. 8. De modulatiediepte m als functie van de spanning V_2 op de elektrode A_2, gemeten met wit licht, in het midden en in de vier hoeken van het beeld. De frequentie was hierbij 3 MHz en de bundelstroom 0,2 μA. Bij $V_2 \approx 67$ V treedt overal het optimale scheidend vermogen op.

antwoord op deze vraag geeft *fig. 8*, waar de modulatiediepte bij een frequentie van 3 MHz is uitgezet, gemeten met wit licht, in het midden en in de vier hoeken van het beeld, als functie van de spanning V_2 op de elektrode A_2. We zien dat bij dezelfde spanning van ongeveer 67 V op alle plaatsen vrijwel de optimale waarde $m = 70\%$ wordt bereikt. In overeenstemming hiermee zijn de met de miniatuurbuizen opgenomen beelden over het gehele oppervlak even scherp.

De fotogeleidende laag

Verkleining van het beeldformaat en wijziging van de dikte van de fotogeleidende laag hebben invloed op de gevoeligheid, het scheidend vermogen en de traagheid van een camerabuis.

De *gevoeligheid* en het *scheidend vermogen* zijn afhankelijk van de golflengte van het licht. De oorzaak hiervan is de absorptie van het licht in de laag. Bij een „Plumbicon" buis wordt licht met een golflengte van 400 nm (blauw) reeds in de eerste microns van de laag geabsorbeerd, terwijl licht van 620 nm (rood) bij een laagdikte van 20 μm slechts voor ongeveer 10% geabsorbeerd wordt. Hierdoor is de buis minder gevoelig voor rood. Daar de laag polykristallijn is, treedt verstrooiing op, en deze doet zich ten gevolge van de geringe absorptie bij rood licht het sterkst gelden, met als gevolg een vermindering van het scheidend vermogen. Verkleining van het beeldformaat moet daarom gepaard gaan met dunner maken van de laag. Doordat dan minder rood licht geabsorbeerd wordt, vermindert echter tegelijkertijd de roodgevoeligheid.

Ontwikkelingen van de laatste tijd stellen ons in staat om zowel het scheidend vermogen als de roodgevoeligheid te verbeteren door de absorptie van de laag voor rood licht te vergroten. Dergelijke lagen zijn reeds eerder toegepast in „Plumbicon" buizen met het standaardformaat[8]. De werkwijze komt erop neer dat men in de loodoxidelaag een zeer geringe hoeveelheid loodsulfide aanbrengt. Deze stof, die ook wordt gebruikt in infraroodfotocellen, absorbeert straling met een golflengte tot 3 μm. Door een geringe concentratie hiervan in de loodoxidelaag wordt de absorptie voor rood licht zodanig vergroot dat de gevoeligheid en het scheidend vermogen veel betere waarden bereiken. Zo kon in de miniatuurbuis het oppervlak van het beeld met een factor 4 verkleind worden en de dikte van de laag met een factor 2. Vergelijkt men de beschreven miniatuur „Plumbicon" buis met een standaardbuis[9], dan vindt men voor de gevoeligheid bij 620 nm waarden van 0,11 respectievelijk 0,04 μA/μW. Wat het scheidend vermogen betreft, valt de vergelijking eveneens gunstig uit: de modulatiediepte bij een frequentie van 5 MHz bedraagt 35% voor de miniatuurbuis (zie boven) en 42% voor de standaardbuis.

De *traagheid* van een „Plumbicon" buis is het meest hinderlijk bij lage lichtniveaus, en de voornaamste bijdrage wordt hierbij geleverd door de bundelstroomtraagheid. Deze traagheid ontstaat doordat de capaciteit van de laag wordt geladen via de bundel, die een

[8] E. F. de Haan, F. M. Klaassen en P. P. M. Schampers, Een experimentele „Plumbicon" opneembuis met verhoogde gevoeligheid voor rood licht, Philips techn. T. **26**, 55-57, 1964/65.

[9] Met standaardbuis is hier bedoeld het type XQ 1020, dat is voorzien van een PbO-laag zoals beschreven is in [7]. Deze buis is algemeen geaccepteerd voor gebruik in studio's.

[10] Zie het aangehaalde artikel [7], blz. 291.

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

zekere weerstand heeft. Hierbij spelen, behalve de capaciteit van de fotogeleidende laag, de signaalstroom, de donkerstroom en de katodetemperatuur een rol. Als op het tijdstip $t = 0$ door een verandering van de verlichtingssterkte de stroom door de laag plotseling van een waarde i_1 op een waarde i_2 wordt gebracht, kan de stroom i_a, die de bundel aan de fotogeleider levert, als functie van de tijd als volgt worden geschreven [10]:

$$i_a(t) = \frac{i_2}{1 + \left(\frac{i_2}{i_1} - 1\right) \exp\left(-\frac{t i_2}{C V_0}\right)}.$$

Hierin is C de capaciteit van de laag en V_0 een constante die evenredig is met de katodetemperatuur. Bij een plotselinge overgang van donker naar licht is i_1 de donkerstroom en $i_s = i_a - i_1$ de signaalstroom; bij een overgang van licht naar donker is i_2 de donkerstroom en $i_s = i_a - i_2$ de signaalstroom.

Uit bovenstaande formule volgt dat bij een verkleining van i_1, i_2 en C met een zelfde factor het relatieve verloop van i_a als functie van de tijd niet verandert. De bundelstroomtraagheid blijft dan ook dezelfde. Nu zou men bij de hierboven genoemde verkleining van het oppervlak en van de dikte van de fotogeleidende laag verwachten dat zowel de capaciteit als de donkerstroom van de miniatuur „Plumbicon" buis tweemaal zo klein zou zijn als bij de standaardbuis. Bij de realisatie van de miniatuurbuis bleek echter dat wèl de capaciteit een factor 2 kleiner was, maar dat de donkerstroom met een factor 3 was toegenomen (1,5 nA tegenover 0,5 nA). De oorzaak hiervan is dat licht van de gloeidraad door de glazen ballon van de buis naar de loodoxidelaag wordt getransporteerd, zodat in de laag, ook als hierop geen beeld wordt geprojecteerd, een zeker verlichtingsniveau aanwezig is. Dit verschijnsel komt in de miniatuurbuis sterker tot uiting dan in de standaardbuis.

In *fig. 9* is op een relatieve schaal de signaalstroom als functie van de tijd getekend voor een plotselinge overgang van donker naar licht. Kromme *a* geldt voor de standaardbuis ($i_1 = 0{,}5$ nA) en kromme *b* voor de miniatuurbuis ($i_1 = 1{,}5$ nA). Bij de berekening van deze krommen is voor de beide buizen niet dezelfde signaalstroom gebruikt. Zoals we nl. bij de bespreking van de camera nog nader zullen zien, is het mogelijk gebleken het ruisniveau zodanig te verlagen dat de miniatuurbuis bij een signaalstroom kan worden gebruikt die ongeveer de helft is van die van de standaardbuis. In verband hiermee is voor de standaardbuis een signaalstroom van 20 nA aangenomen en voor de miniatuurbuis een signaalstroom van 10 nA. Uit fig. 9 blijkt dat de miniatuurbuis een geringere traagheid heeft dan de standaardbuis. Dit wordt ten dele veroorzaakt door de kleinere capaciteit en ten dele door de grotere donkerstroom.

In *fig. 10* zijn de resultaten weergegeven van overeenkomstige berekeningen bij een plotselinge overgang van licht naar donker. Nog meer dan uit fig. 9 blijkt hier

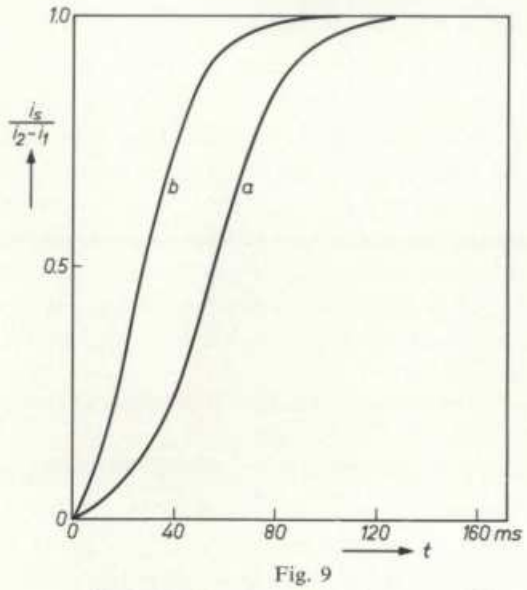


Fig. 9

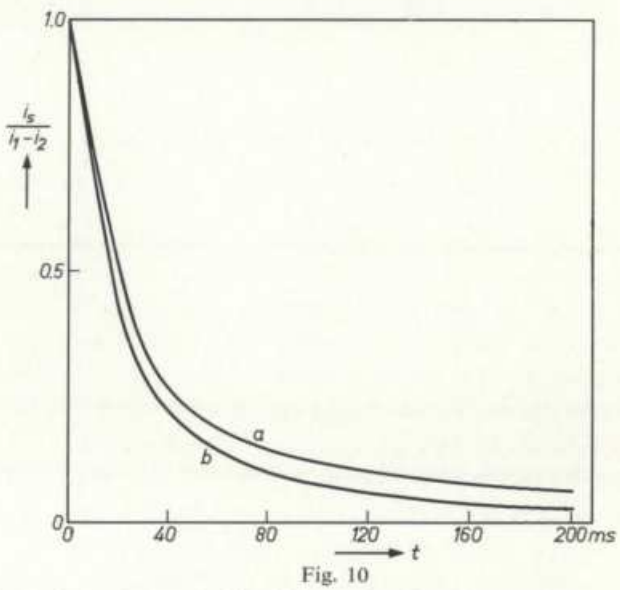


Fig. 10

Fig. 9. De signaalstroom i_s (op een relatieve schaal) als functie van de tijd t bij een plotselinge overgang van donker naar licht. De stroom door de fotogeleidende laag verandert van i_1 (donkerstroom) naar i_2. De kromme *a* geldt voor de standaard „Plumbicon" buis (donkerstroom 0,5 nA, capaciteit van de fotogeleidende laag 1 nF), waarbij als uiteindelijke waarde van de signaalstroom 20 nA is aangenomen. De kromme *b* geldt voor de miniatuurbuis (donkerstroom 1,5 nA, capaciteit 0,5 nF), bij een uiteindelijke signaalstroom van 10 nA.

Fig. 10. De signaalstroom i_s (op een relatieve schaal) als functie van de tijd t bij een plotselinge overgang van licht naar donker, *a* voor de standaard „Plumbicon" buis, *b* voor de miniatuurbuis.

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

de geringere traagheid van de miniatuur „Plumbicon" buis. (Speciaal in het onderste gedeelte van de figuur is de horizontale afstand van de krommen groot.)

De camera

Bij de hier beschreven eerste ontwikkelingsfaze van de kleine camera wilden wij gebruik maken van het controlegedeelte van de studiocamera [4]. Om de afmetingen van het eigenlijke cameragedeelte zoveel mogelijk te beperken, werd echter besloten om hierin alleen dat deel van de schakeling te plaatsen dat zich vlak bij de opneembuizen moet bevinden. De rest werd ondergebracht in een afzonderlijke kast, de „tussenkast", die via een korte kabel met het cameradeel is verbonden (zie *fig. 11*). Deze tussenkast is dan verder weer aangesloten op het normale controlerek.

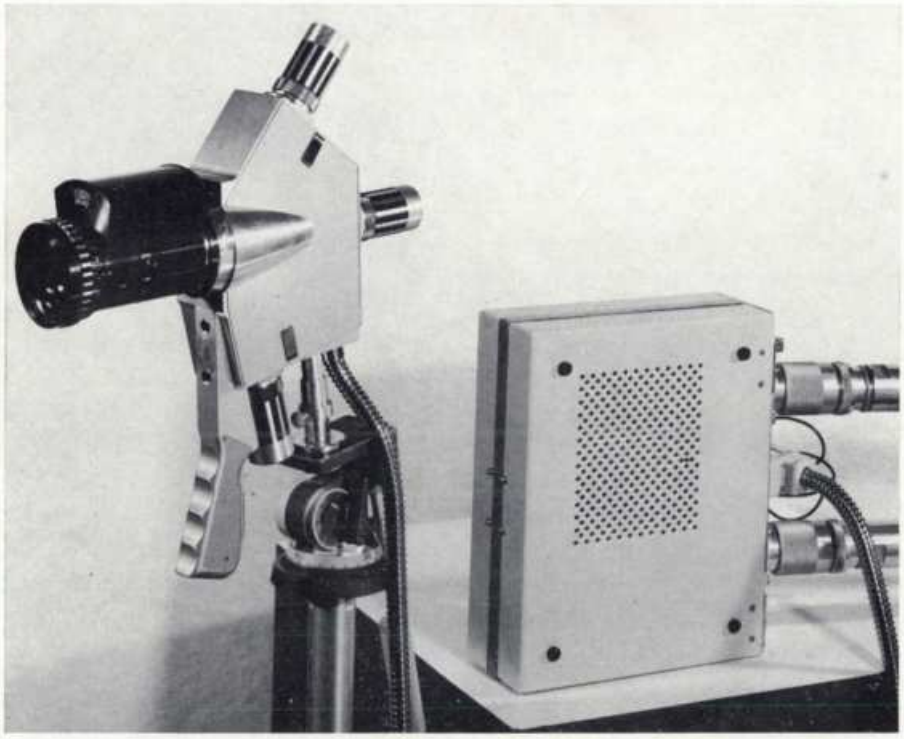


Fig. 11. De miniatuurcamera met de „tussenkast". Afmetingen van deze kast 18 × 24 × 10 cm.

Fig. 12 toont één zijde van de geopende camera. Men ziet het kleurscheidingsprisma en de drie opneembuizen met hun deflexiespoelen. Aan de andere kant bevinden zich de drie voorversterkers, de versterker voor de bundelonderdrukkingsimpulsen en andere schakelingen die nodig zijn voor de werking van de „Plumbicon" buizen.

Het optische systeem

Het optische gedeelte van de camera is in principe gelijk aan dat van de studiocamera. Ook hier zijn de drie opneembuizen niet evenwijdig geplaatst. De dan noodzakelijke afscherming tegen uitwendige velden is bij de kleine afmetingen gemakkelijker uit te voeren dan bij de grote camera.

Het *objectief* is speciaal voor dit kleine beeldformaat ontwikkeld; de brandpuntsafstand kan worden gevarieerd van 13 tot 65 mm (zoomlens). De relatieve opening is f : 1,6. Door het kleinere beeldformaat is bij de grootste opening de scherptediepte even groot als bij de studiocamera, die is uitgerust met een lens met relatieve opening f : 3,2. Het kleurscheidingsprisma is aangepast aan de kleine afmetingen en aan de grote lensopening; de hoeken die de spiegelende oppervlakken met de optische as maken zijn iets groter gekozen, om afsnijden van de randen van het beeld (vignettering) te voorkomen.

De superpositie van de beelden

In de door de kleurscheiding verkregen beelden op de drie „Plumbicon" buizen moeten de punten die corresponderen met een zelfde punt van de opgenomen scène gelijktijdig worden afgetast. De nauwkeurigheid waarmee dit geschiedt hangt van vele factoren af. De eisen die hiervoor aan de nauwkeurigheid en de stabiliteit van de mechanische constructie worden gesteld, zijn bij dit kleine beeldformaat zwaarder dan bij de standaardbuizen in de studiocamera. Om deze reden is het

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

camerahuis uit één stuk gefreesd. De deflexiespoelen, die op onderlinge gelijkheid zijn uitgezocht, kunnen, evenals in de grote camera, door middel van stelschroeven nauwkeurig in de juiste stand ten opzichte van de optische as worden gebracht. Om hierbij onafhankelijk te zijn van de toleranties in de constructie van de buizen, worden bij deze instelprocedure in plaats van de „Plumbicon" buizen op maat gedraaide pijpen van methylmethacrylaat gebruikt. De hierna in de camera geplaatste „Plumbicon" buizen kunnen eveneens door middel van stelschroeven worden verplaatst, zodat men hun elektronenoptische assen kan laten samenvallen met de assen van het optische systeem. Zou men dit niet doen, dan zouden ten gevolge van onderlinge geometrische verschillen in de drie buizen moeilijkheden ontstaan bij het gelijktijdig aftasten van de drie beelden. Bovendien zouden de drie beelden ten opzichte van elkaar verschoven kunnen zijn, waardoor een minder effectief gebruik zou worden gemaakt van het oppervlak van de gevoelige laag.

Van de verdere maatregelen die zijn genomen om de superpositie van de beelden te bevorderen noemen wij nog het feit dat de verschillen in impedantie van de (parallelgeschakelde) deflexiespoelen van de drie buizen met variabele zelfinducties en weerstanden zijn gecorrigeerd.

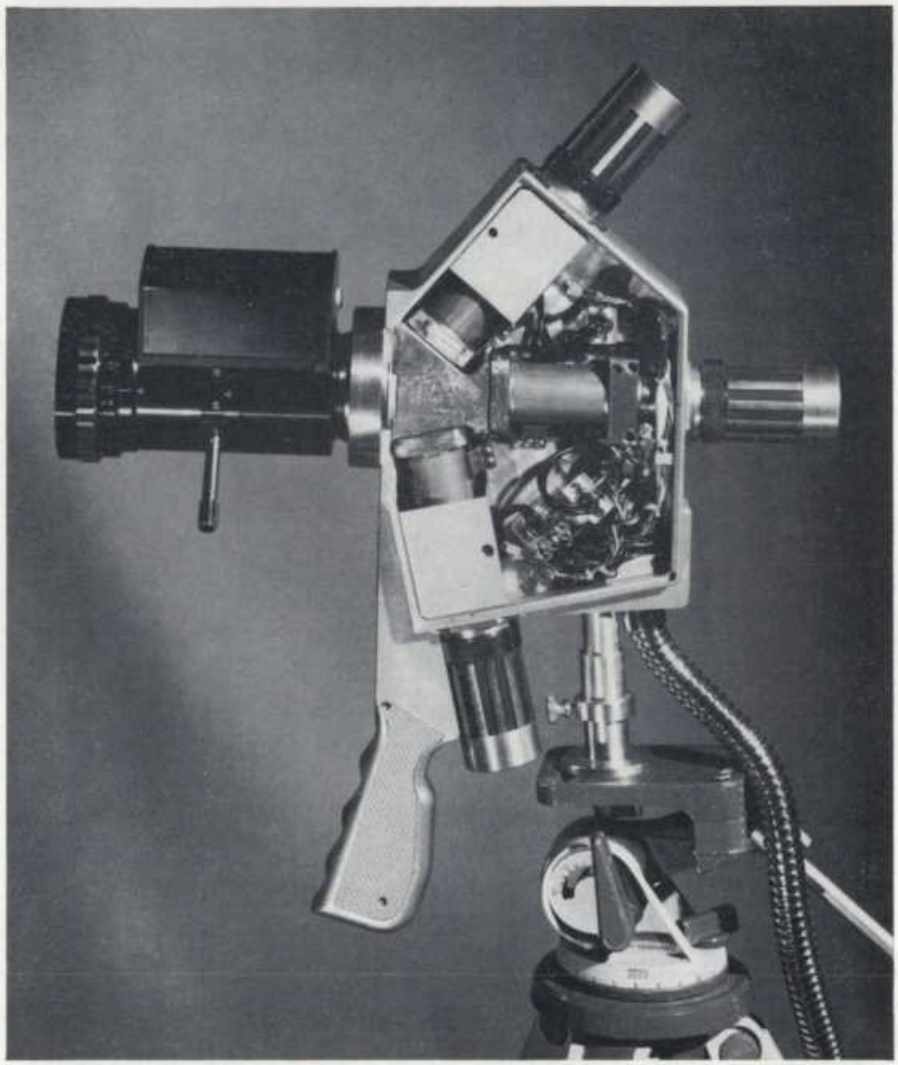


Fig. 12. De geopende camera. Men ziet het kleurscheidingsprisma en de drie „Plumbicon" buizen met hun deflexiespoelen. Aan de andere zijde zijn enkele elektronische schakelingen ondergebracht.

De gevoeligheid

De gevoeligheid van een „Plumbicon" camera wordt enerzijds begrensd door de toelaatbare traagheid bij het opnemen van bewegende beelden en anderzijds door de ruis. Wanneer bij de studiocamera een veldeffect-

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

transistor wordt gebruikt in de ingangstrap van de signaalversterker, is de ruis zo gering dat in de praktijk de traagheid bepalend is voor de gevoeligheid. Bij de miniatuur „Plumbicon" buizen is echter de traagheid zoveel kleiner dat hier de begrenzing vrijwel uitsluitend door de ruis gevormd wordt.

Voor een grote signaal-ruisverhouding moet de ingangsimpedantie van de signaalversterker groot zijn.

Gebruikt men een camera in combinatie met andere optische instrumenten, dan spelen de problemen van de onderlinge koppeling een belangrijke rol. Hierbij treedt nl. vaak een groot verlies in lichtsterkte, dus in gevoeligheid, op. Zo is men b.v. bij het combineren van een endoscoop en een grote camera gedwongen een flexibele optische koppeling toe te passen, met het genoemde gevolg. Het feit dat de miniatuurcamera *direct* op een

Fig. 13. De eerder beschreven studiocamera [4] en de miniatuurcamera, beide uitgerust met „Plumbicon" buizen. (Bij de miniatuurcamera behoort nog de in fig. 11 afgebeelde „tussenkast" die hier niet zichtbaar is.) Ter vergelijking van de beeldkwaliteit wordt hier met beide camera's hetzelfde object opgenomen. Met behulp van een elektronische schakelaar worden op de rechts zichtbare monitor de beelden van beide camera's gelijktijdig zichtbaar gemaakt. Het is helaas niet mogelijk om het monitorbeeld en de rest van de opstelling te zamen op kleurenfilm van een bepaalde soort vast te leggen zonder dat er afwijkingen in de kleurbalans optreden. Het monitorbeeld is in fig. 14 gereproduceerd.

Deze impedantie wordt gevormd door de parallelschakeling van de signaalweerstand en de capaciteiten van de signaalplaat, van de versterker en van de bedrading. Door de kleine afmetingen en door een uiterst korte verbinding tussen signaalplaat en versterker kon de totale capaciteit kleiner dan 10 pF worden gehouden. Hierdoor, en ook door het feit dat als ingangstrap van de versterker een veldeffecttransistor wordt gebruikt met bijzonder gunstige ruiseigenschappen, werd bij een zelfde signaalstroom een 4 à 6 dB grotere signaal-ruisverhouding bereikt dan bij de studiocamera.

Mede door de betere traagheidseigenschappen kunnen de buizen in de kleine camera dan ook met een twee- à driemaal kleinere signaalstroom worden gebruikt dan de standaard „Plumbicon" buizen in de grote camera. Bij de maximale opening van de lens is een scèneverlichting van 400 à 500 lux voldoende.

endoscoop kan worden aangesloten geeft hier een belangrijke winst in gevoeligheid.

De beeldscherpte

Door het goede scheidend vermogen van de miniatuur „Plumbicon" buizen is de scherpte over het gehele beeldvlak verrassend goed. Een vergelijking van de studiocamera en de miniatuurcamera is te maken aan de hand van een experiment dat is geïllustreerd in *fig. 13* en *fig. 14*. Evenals bij de studiocamera kan men de beeldcontouren nog verscherpen door het gebruik van contoursignalen, afgeleid uit het groene kleursignaal [11].

Door de geringere traagheid van de miniatuur „Plumbicon" buis is zelfs, bij de kleinere signaalstroom, de scherpte van bewegende beelden beter dan bij de studiocamera.

Maakt men gebruik van de bovengenoemde moge-

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

lijkheid om de miniatuurcamera direct te koppelen met andere optische instrumenten, b.v. een microscoop, dan kan men nog een verdere verbetering in scherpte verkrijgen dank zij het feit dat de hierbij op te nemen beelden veelal rond zijn. Men kan dan nl. de deflexiestromen zodanig vergroten dat de gehele, cirkelvormige, fotogeleidende laag wordt gebruikt. Door het feit dat men dank zij deze vergroting van het beeldformaat met minder contourencorrectie kan volstaan, wordt dan tevens de signaal-ruisverhouding nog verbeterd. (Natuurlijk is deze vergroting van het beeldformaat eveneens van invloed op de gevoeligheid en de traagheid.)

De resultaten die zijn verkregen met de experimentele camera zijn reeds zodanig dat men deze camera met voordeel zou kunnen gebruiken wanneer inbouw, ophanging of koppeling van een grote camera moeilijkheden oplevert. Verschillende problemen, zowel betrekking hebbende op de opneembuizen als op de camera, vragen echter nog om een oplossing. Zo is het, om de voordelen van een draagbare camera voor reportagedoeleinden ten volle te kunnen benutten, gewenst het deel van de elektronische schakelingen dat zich in de tussenkast bevindt in het camerahuis onder te brengen. Dit levert dan een compleet kleurensignaal en het kan door middel van een dunne kabel met de reportagewagen worden verbonden. Aan een dergelijke camera wordt thans gewerkt.

Fig. 14. Scherm van de monitor in de opstelling van fig. 13. Op de linkerhelft een deel van het object opgenomen met de studiocamera, die voor dit experiment was uitgerust met „Plumbicon" buizen met verhoogde roodgevoeligheid; op de rechterhelft een overeenkomstig deel van hetzelfde object opgenomen met de miniatuurcamera.

[11] Zie het aangehaalde artikel [4], blz. 356 e.v.

Samenvatting. Beschrijving van een experimentele kleine kleurentelevisiecamera, die is uitgerust met hiervoor ontwikkelde miniatuur „Plumbicon" opneembuizen met elektrostatische focussering. De afmetingen van het beeld op de fotogeleidende laag zijn hierbij slechts 8,4×6,3 mm. Ondanks deze kleine afmetingen wordt een beeldkwaliteit verkregen die een vergelijking met die van de beelden van een studiocamera goed kan doorstaan. Behalve voor gebruik in studio's en bij reportages biedt deze camera ook grote voordelen als men beelden wil weergeven die met een ander optisch instrument, b.v. een microscoop, worden opgenomen. Men kan dan nl. deze kleine camera direct met de desbetreffende instrumenten koppelen. Het verlies in gevoeligheid en in beeldscherpte dat bij een grote camera optreedt door de noodzakelijke koppelende elementen wordt dan verminderd.

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

Het opstellen van computerprogramma's

L. Crousel

Met het tot rijpheid komen van de computers wordt de aandacht hoe langer hoe meer gericht op de manier waarop men deze het best kan gebruiken. Voor het gros van de gebruikers van computers heeft de belangstelling zich de laatste jaren verschoven van de „machinecode" via „symbolische talen" naar de zg. „hogere programmeertalen". Deze laatste hebben het voordeel dat ze meer zijn aangepast aan de problemen die ze moeten beschrijven, dat ze onafhankelijk zijn van speciale machines, en dat ze de programmeur zoveel mogelijk programma-organisatie uit handen nemen. Bovendien blijken ze vaak een uitstekend middel te vormen voor schriftelijke communicatie tussen vakgenoten over toegepaste algoritmes. In het hier volgende artikel bespreekt de auteur de belangrijkste stadia bij het opzetten van programma's voor wetenschappelijke problemen, waarbij hij aan de hand van eenvoudige voorbeelden enkele kenmerken van „hogere programmeertalen", „symbolische talen" en „machinecodes" en hun samenhang belicht.

De tegenwoordige ontwikkeling van computersystemen maakt deze systemen voor de gebruiker veel beter toegankelijk: in zijn werkkamer wordt apparatuur opgesteld voor directe verbinding met de computer door toetsenbord of beeldscherm; standaardprogramma's voor vaak terugkerende ontwerp-problemen komen beschikbaar om ontwerpers bij hun werk te helpen; hogere programmeertalen worden ontwikkeld om algoritmes op overzichtelijke manier uit te drukken. Om van deze ontwikkelingen te kunnen profiteren, is het noodzakelijk dat onderzoekers en ingenieurs op de hoogte zijn van de diensten die de rekencentra kunnen verlenen. Dit houdt onder meer in dat zij zich met de directe invoer in de computer vertrouwd maken en dat zij de standaardprogramma's leren gebruiken. Tevens raken de rekencentra steeds meer belast met de algemene programmatuur („basic software"), zodat voor het schrijven van applicatieprogramma's voor speciale problemen de gebruikers steeds meer op zichzelf aangewezen zijn. Enkele richtlijnen hiervoor zullen in dit artikel worden uitgestippeld. Een dergelijke methodologische inleiding lijkt ons niet ondienstig, aangezien de te volgen gedachtegang bij het opstellen van een programma soms zeer ingewikkeld kan zijn.

Indeling van het proces

Het is vaak helemaal niet eenvoudig te zeggen of de uitkomsten die een computer bij het uitvoeren van een programma levert, wel zinvol zijn. Het toetsen hiervan kan zelfs tot diepgaande discussies leiden, omdat op allerlei punten twijfel kan bestaan: het mathematische model kan ontoereikend zijn, of de rekenmethode ongeschikt, een programma kan nog andere effecten hebben dan de bedoelde, of de computer zelf kan fouten maken. Ten einde niet achteraf voor het haast onoplosbare probleem te staan om na te gaan waar de schuld ligt, dient men een duidelijke scheiding van een aantal taken in te voeren. Dit levert nog andere belangrijke voordelen op: men kan de taken eventueel over verschillende personen verdelen; door concentratie op één der taken kan deze beter worden doordacht en uitgevoerd; bovendien stimuleert het de vindingrijkheid en bevordert het de afzonderlijke controle van elk onderdeel.

Het splitsen in verschillende taken behoeft niet te betekenen dat de grenzen hiertussen scherp en onveranderlijk zijn. Veeleer is een goede uitwisseling van gedachten over deze grenzen heen essentieel om het uiteindelijke doel te bereiken, maar de gedachtewisseling kan slechts effectief zijn als duidelijk vaststaat waarover men het heeft.

Taak 1 is de probleemanalyse, die moet resulteren in een programmaspecificatie, d.i. een precieze opgave van wat de invoergegevens zijn en wat men als resultaat verwacht.

Taak 2 is het opzetten van een rekenschema of algoritme, d.i. een reeks van opeenvolgende bewerkingen, waarmee het door taak 1 gedefineerde doel wordt gerealiseerd.

L. Crousel, medewerker van MBLE Laboratoire de Recherches, Brussel.

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

Taak 3 is het eigenlijke programmeren: het schrijven van deze algoritme in een taal waarmee de computerapparatuur kan werken.

Taak 1: de probleemomschrijving en de programmaspecificatie

Heeft men een bepaald vraagstuk voor welks oplossing men een programma wil opstellen, dan moet men om te beginnen alle nodige informatie verzamelen, een analytische formulering zoeken, zich rekenschap geven van de parameters die de uitkomst beïnvloeden en van de grootheden die men als resultaat wenst te kennen, en men moet de algemene gedachtegang voor de oplossing schetsen. Dit alles vat men soms samen onder de naam „*systeemanalyse*" [1], naar analogie met het begrip „systeemontwerp" bij de ontwikkeling van apparatuur [2]. Het doel is een volledige omschrijving van wat computer èn programma samen moeten doen.

Bij het oplossen van een vraagstuk komt *altijd* een systeemanalyse te pas, ook al wil men geen computer gebruiken. De computer kan er niet bij helpen, en hij maakt dit deel van de opgave alleen nog moeilijker dan normaal, daar het bij het automatisch rekenen vrijwel onmogelijk is om telkens tussenresultaten te controleren. Ziet men bij de analyse iets over het hoofd, dat doorwerkt in het programma, dan kan dit later bij de uitvoering door de rekenmachine onopgemerkt blijven en tot foutieve resultaten leiden, terwijl men bij oplossing langs normale weg de fout ontdekt zou hebben. Een typisch voorbeeld vormt het geval van het quotiënt van twee functies, dat voor zekere waarden van de argumenten de onbepaalde vorm 0/0 zou aannemen. Heeft men dit over het hoofd gezien, dan is de kans groot dat beide functies voor de betrokken waarden van de argumenten door nauwkeurigheidsverlies iets van nul verschillende kleine waarden krijgen, waardoor het quotiënt nu — ongemerkt — volledig fout kan zijn.

Veelal is de systeemanalyse de belangrijkste opgave bij het opstellen van een programma. Toch is het nauwelijks mogelijk een overzicht te geven van alle stappen die erbij te pas komen, wegens de grote verscheidenheid van problemen die voor oplossing met de computer in aanmerking komen. De volgende beschouwing is derhalve zeker niet volledig en evenmin voor alle gevallen van toepassing.

De behandeling van problemen in uiteenlopende gebieden van wetenschap (natuurkunde, scheikunde, techniek, economie enz.) komt erop neer dat men begrippen door variabelen karakteriseert en deze door wiskundige betrekkingen met elkaar verbindt. Dit noemt men een wiskundig *model*, en hiervan moet men uitgaan voor verder onderzoek en numerieke berekening. Stel b.v. dat we de elektronenbanen in een katodestraalbuis, voor een bepaalde instelling van de buis, willen berekenen [3]. Als we het buistype opgeven, is het vraagstuk reeds ondubbelzinnig vastgelegd. Voor het opstellen van een model moeten nu de variabelen gedefinieerd worden die erbij te pas komen (zoals potentiaal, veldsterkte, krachten enz.), terwijl men de fysische en mechanische wetten die het verschijnsel beheersen moet uitdrukken b.v. in de vorm van differentiaalvergelijkingen.

De systeemanalyse is echter met een wiskundige beschrijving van het vraagstuk nog niet klaar. Er moet ook een gedachtegang, een soort geraamte voor de oplosmethode, worden gegeven welke dan in taak 2 nader wordt uitgewerkt. Daarbij kan het hele arsenaal van algebra, differentiaal- en integraalrekening, numerieke benaderingsmethodes enz. te pas worden gebracht, maar men moet bij de keuze van de methode rekening houden met de eigenaardigheden van het automatisch rekenen. Een der aspecten is het afwegen van rekentijd tegen programmeertijd. Om b.v. het minimum van een functie in een zeker geval te bepalen, kan het beter zijn de functie (met een regelbare stapgrootte) tabellarisch te berekenen, dan een methode toe te passen waarmee men het nulpunt van de afgeleide bepaalt. In het eerste geval kan nl. het programmeren veel minder tijd vereisen, terwijl het verschil in rekentijd misschien nauwelijks merkbaar is. Een ander aspect van het rekenen met de computer is dat men profijt kan trekken van vroeger gedaan werk. Een rekencentrum bezit meestal een verzameling standaardprogramma's voor het oplossen van veel voorkomende vraagstukken. Zulke vraagstukken zijn b.v. het berekenen van allerlei functies, het oplossen van stelsels lineaire vergelijkingen, het inverteren van matrices, diverse procédés voor numerieke integratie, enz. Dergelijke programma's vormen bouwstenen waarmee grotere programma's kunnen worden samengesteld.

Bekendheid met de eigenaardigheden en mogelijkheden van het machinaal rekenen is dus zeer nuttig als men oplossingen met behulp van de computer wil uitwerken. Dit betekent echter niet dat de systeemanalist geen steun zou behoeven van de onderzoeker die het wiskundig model heeft opgesteld. Kennis van de theorie die aan het model ten grondslag ligt kan hem vaak op een idee brengen voor mogelijke oplosmethodes die uit de puur wiskundige analyse niet blijken.

Samenvattend kan men zeggen dat het gebruik van een computer voor de oplossing van vraagstukken wel

[1] C. Strachey, System analysis and programming, Sci. Amer. **215**, no. 3, 112-124, 1966.

[2] G. M. Amdahl, G. A. Blaauw en F. P. Brooks, Jr., Architecture of the IBM System/360, IBM J. Res. Devel. **8**, 87-101, 1964.

[3] C. Weber, Het berekenen van potentiaalvelden en elektronenbanen met behulp van een elektronische rekenmachine, Philips techn. T. **24**, 130-143, 1962.

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

een aanzienlijke arbeidsbesparing kan opleveren, maar zeker geen besparing in denkwerk. De programmaspecificatie moet daarbij uiterst nauwkeurig worden opgesteld en dit vereist een grondige systeemanalyse. Werkt men met een tafelrekenmachine of een rekenlineaal, dan kan men gaandeweg het probleem doordenken en van koers veranderen als er delen van de oplossing in zicht komen. Bij de computer kan dit niet zo gemakkelijk. Weliswaar is het mogelijk en soms zelfs wenselijk om het vraagstuk in stukken te verdelen en eerst de deelresultaten te bekijken, maar voor ieder deelvraagstuk heeft men dan dezelfde strenge eisen.

De programmaspecificatie is het uitgangspunt voor het opstellen van het rekenschema. Als zodanig is deze omschrijving gewoonlijk de basis voor het contact tussen de steller en de oplosser van het probleem, wanneer b.v. het eigenlijke programmeren door het rekencentrum wordt verricht. Dit betekent wederom niet dat de oplosser de steller geheel moet volgen, want bij de uitvoering kan weleens blijken dat het dienstig is de specificatie te veranderen. Zo kan het voorkomen dat de programmaspecificatie verscheidene malen gewijzigd wordt totdat beide partijen het over de geschiktheid en volledigheid ervan eens zijn.

Taak 2: het opstellen van het rekenschema (de algoritme)

Heeft men een goede programmaspecificatie, dan kan men aan de taak beginnen een rekenschema voor het oplossen met de computer op te stellen. De analist die deze taak uitvoert moet in staat zijn een strategie te ontwerpen die steunt op alle beschikbare hulpmiddelen (waaronder de gangbare numerieke benaderingsmethodes) en die past bij de eigenaardigheden van de computer.

Ter illustratie van de voorwaarden die door de structuur van de computer worden opgelegd, beschouwen wij het zg. „probleem van de handelsreiziger": zoek de kortste weg om, uitgaande van een vast beginpunt, m gegeven plaatsen éénmaal aan te doen en op het beginpunt terug te keren. De meest voor de hand liggende aanpak bestaat daarin, successievelijk alle denkbare wegen te beschouwen, hun lengte te berekenen en de kortste daaruit te kiezen. Dit vereist een methode voor het genereren van alle permutaties van m elementen (de m plaatsen). Hun aantal is P_m, met de recurrente formule:

$$P_m = m\,P_{m-1}.$$

Het genereren van de permutaties kan geschieden met dezelfde iteratieve methode die men gewoonlijk gebruikt om de bovenstaande formule te bewijzen. Daarbij is het echter nodig $(m-1)!$ permutaties van $m-1$ elementen te onthouden. Bij grote m wordt dit al spoedig ondoenlijk. Men moet naar een betere strategie zoeken, met name naar een methode die alle permutaties successievelijk oplevert zonder dat een groot aantal tevoren gevormde permutaties onthouden behoeft te worden. Inderdaad bestaan er methodes waarbij men behalve een eenvoudige boekhouding (een aantal tellingen) alleen de laatste gevormde permutatie nodig heeft. Zo is er een door Landon aangegeven methode, gebaseerd op cyclische permutaties van delen van de reeks elementen [4]. Het typische is dat men deze eenvoudige methode pas bedacht heeft toen men met het probleem van de programmering voor de computer geconfronteerd werd.

Als tweede voorbeeld beschouwen wij het tabelleren van de functie

$$A(\gamma) = \ln \left| \prod_{i=1}^{n} \frac{e^{\gamma}/e^{\gamma_i} + e^{\gamma_i}/e^{\gamma}}{e^{\gamma}/e^{\gamma_i} - e^{\gamma_i}/e^{\gamma}} \right|, \quad \ldots \quad (1)$$

die bij het ontwerp van filters wordt gebruikt. Met de hulpgrootheden $w_i = e^{\gamma_i}$ en $v = e^{\gamma}$ heeft men:

$$A(\gamma) = \ln \left| \prod_{i=1}^{n} \frac{v/w_i + w_i/v}{v/w_i - w_i/v} \right|. \quad \ldots \quad (2)$$

De w_i zijn hier constanten, die tevoren kunnen worden berekend. Het computerprogramma dat van formule (2) uitgaat vergt behalve de elementaire operaties van optellen, aftrekken, vermenigvuldigen en delen, die naar verhouding weinig tijd kosten, slechts het berekenen van één e-macht (e^{γ}) en één logaritme voor elke tabelleerstap γ. Nu wordt in de praktijk de functie (1) gewoonlijk geschreven als

$$A(\gamma) = \sum_{i=1}^{n} \ln \coth |(\gamma - \gamma_i)|. \quad \ldots \quad (3)$$

In het computerloze tijdperk was deze uitdrukking heel handig: heeft men een tabel van $\ln \coth |x|$ ter beschikking, dan hoeft men om de functie voor een bepaalde γ-waarde te berekenen slechts n maal in de tabel te zoeken en n optellingen te doen. Bij het automatisch rekenen echter betekent het opslaan van functies in tabelvorm verkwisting van geheugenruimte. Men zou derhalve de genoemde functie nadat men ze tot logaritmes en e-machten heeft teruggebracht uit de reeksontwikkelingen hiervoor moeten berekenen. Daar het aantal standaardfuncties dat men voor formule (3) berekenen moet n maal zo groot is als voor formule (1) (nl. n logaritmes en n hyperbolische functies, die neerkomen op e-machten) zou ook de rekentijd bij toepassing van formule (3) vrijwel n maal zo groot zijn als bij formule (1). Deze prijs zou men dus betalen als men niet lette op de eigenaardigheden van het automatisch rekenen.

De gegeven voorbeelden illustreren hoe bekendheid met de eigenaardigheden van het automatisch rekenen richtlijnen oplevert voor het opstellen van reken-

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

schema's, al blijft dit een proces met vele vrijheidsgraden, waarbij het initiatief en de achtergrondkennis van de analist terdege hun rol spelen. Het uitstippelen van de rekenmethode levert echter nog geen vaste basis voor het schrijven van het programma. Wat men uiteindelijk nodig heeft, is een gedetailleerde en ondubbelzinnige beschrijving van de reeks bewerkingen die uitgevoerd moeten worden om een bepaald resultaat te bereiken. Hiertoe is een formalisme ontwikkeld dat zijn neerslag vindt in het *blokschema* (*flow chart*).

Het blokschema heeft de belangrijke eigenschap dat het ondubbelzinnig de volgorde vastlegt waarin bewerkingen moeten worden uitgevoerd. Bij de gebruikelijke voorstellingswijze schrijft men elke bewerking in een blokje, en de opeenvolging wordt vastgelegd door pijlen tussen de blokken. *Fig. 1* geeft een eenvoudig voorbeeld, het blokschema van een methode (algoritme) voor het berekenen van y, de absolute waarde van x. Het schema bevat drie bewerkingen, waarvan de eerste een vertakking behelst: er wordt een vraag gesteld, en naar gelang het antwoord uitvalt, volgt men de vertakking gemerkt „ja" of „neen".

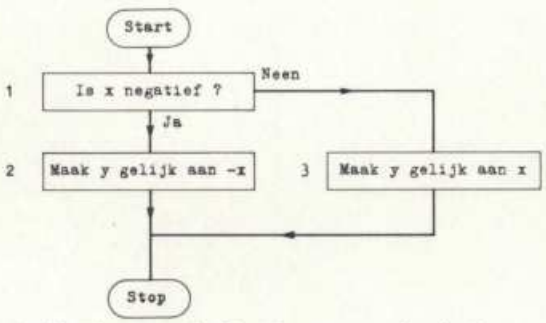


Fig. 1. Algoritme voor het berekenen van y, de absolute waarde van een gegeven variabele x.

De bewerking binnen een blok kan een probleem op zichzelf voorstellen, waarvan de oplossing op zijn beurt in een blokschema beschreven is. Door deze trapsgewijze omschrijving van het gehele rekenschema wordt het probleem in deelproblemen gesplitst en de omvang en ingewikkeldheid van elk stuk waarmee men op een bepaald moment te maken heeft beperkt. Het blokschema in de programmatuur (software) komt geheel overeen met dat in de apparatuur (hardware).

Blokschema's bevorderen de logische opzet doordat zij de aandacht concentreren op de onderlinge samenhang van de blokken en een vergelijking van verschillende opzetten reeds in een vroeg stadium mogelijk maken. Ter illustratie hiervan beschouwen wij de berekening van een matrix B die gelijk is aan de n-de macht van een vierkante matrix A (n is een geheel getal). In het blokschema van *fig. 2* is een mogelijke werkwijze aangegeven. Bewerking 6 is de berekening

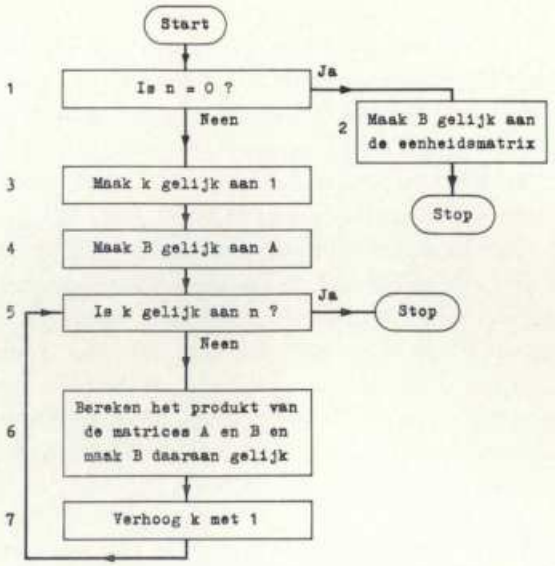


Fig. 2. Algoritme voor het berekenen van B, de n-de macht van een vierkante matrix A.

van het produkt van twee matrices, een vraagstuk waarvan de oplossing vervat is in een blokschema op een lagere trap en waarvan de details hier niet ter zake doen. Het schema voldoet aan de eis dat voor $n = 0$ de eenheidsmatrix als resultaat verkregen moet worden; hiervoor zorgt de vertakking 1 aan het begin. Aan deze eis hadden we ook kunnen voldoen zonder de vertakking, door de opzet van het schema iets te wijzigen: de kern van het schema, bestaande uit de matrixvermenigvuldiging blijft dezelfde, maar de bewerkingen 3 en 4 worden vervangen door:

3 Maak k gelijk aan 0

4 Maak B gelijk aan de eenheidsmatrix.

Het nadeel van deze andere opzet is nu meteen te zien: voor alle $n \neq 0$ duurt het proces één matrixvermenigvuldiging langer.

Uit de gegeven voorbeelden blijkt hoe gewenst het is een beknopte notatie voor het aangeven van afzonderlijke bewerkingen bij de beschrijving van rekenschema's in te voeren. De bewerking 3 in fig. 1 is de toekenning van een waarde aan een variabele en zal worden voorgesteld door de formule:

$$y := x,$$

een notatie die is ontleend aan de programmeertaal ALGOL 60. (Te lezen: y wordt gelijk aan x.) De operator $:=$ is zo gekozen om de aandacht te vestigen op de asymmetrie van de bewerking. Deze komt in de gewone notatie

$$y = x$$

niet tot uitdrukking, daar deze notatie alleen een verband tussen twee variabelen aangeeft zonder te ver-

[4] G. G. Langdon, Jr., An algorithm for generating permutations, Comm. ACM **10**, 298-299, 1967 (no. 5).

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

melden dat de ene moet worden verkregen uit de andere. De door het toekenningsvoorschrift (assignment statement) $y := x$ aangeduide bewerking houdt in dat aan de variabele met het symbool y een waarde (de waarde van x) wordt toegekend, hetgeen men tijdens het rekenprocédé naar believen met steeds andere waarden kan doen. We drukken dit uit door te zeggen dat zulke variabelen uit een rekenschema *dynamisch* zijn. In tegenstelling hiermee noemen we de variabelen in een wiskundige betrekking statisch, aangezien de toevoeging van hun waarden aan die van andere variabelen, zoals uitgedrukt door de betrekkingen, binnen het kader van de gehele wiskundige beschrijving onveranderd blijft. Bezien wij b.v. de iteratieformule van Newton-Raphson:

$$x_{n+1} = x_n - \mathrm{f}(x_n)/\mathrm{f}'(x_n), \quad \ldots \quad (4)$$

die voor het verbeteren van de benadering x_n van een nulpunt van de functie $\mathrm{f}(x)$ dient. De algoritme voor het uitvoeren van deze regel wordt uitgedrukt door de notatie:

$$x := x - \mathrm{f}(x)/\mathrm{f}'(x). \quad \ldots \quad (5)$$

In verg. (4) worden opeenvolgende benaderingen belichaamd door een reeks statische variabelen, terwijl in het voorschrift (5) deze benaderende waarden successievelijk aan een enkele dynamische variabele x worden toegekend.

Evenzo bewerkstelligt het toekenningsvoorschrift

$$i := i + 1$$

dat de variabele i elke keer als het voorschrift wordt uitgevoerd met een eenheid wordt verhoogd. Een dergelijke bewerking wordt vaak gebruikt om een telling bij te houden. Deze moet dan wel worden voorafgegaan door een voorschrift dat aan i een beginwaarde toekent, b.v. $i := 0$.

Door het dynamische karakter van de variabelen in het blokschema bespaart men symbolen, en derhalve, zoals we later zullen zien, ook geheugenplaatsen. Anderzijds is het de belangrijkste oorzaak waarom het beschrijven of bespreken van een rekenmethode zo moeilijk is, vooral als een zelfde variabele voor allerlei doeleinden binnen het rekenschema wordt gebruikt.

Naar gelang de behoefte zich voordoet, kan men eigen notaties invoeren, maar in de meeste gevallen komt men met min of meer gestandaardiseerde uit. Zo stelt men de in fig. 1 voorkomende vertakkingsoperatie 1 voor zoals in *fig. 3* is te zien. Door de ruitvorm van het blokje is deze operatie duidelijk te onderscheiden van bewerkingen die géén invloed hebben op de volgorde.

We gaan nu het opzetten van een rekenschema iets uitvoeriger bekijken en beschouwen daartoe als voorbeeld de berekening van het produkt C van twee matrices A en B; in formule:

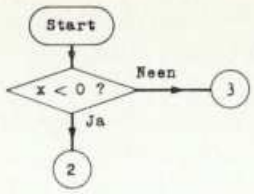


Fig. 3. Symbool voor een vertakking in een blokschema.

$$C^p_m = A^n_m \, B^p_n .$$

De indices onderaan en bovenaan geven resp. het aantal rijen en kolommen van de desbetreffende matrix (zie *fig. 4*). Het element c_{ij} in rij i en kolom j van de produktmatrix wordt uit de, overeenkomstig aangeduide, elementen a_{ik} van de matrix A en b_{kj} van de matrix B berekend volgens de betrekking:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} , \quad \ldots \quad (6)$$

met $i = 1, 2, \ldots m,$
$j = 1, 2, \ldots p.$

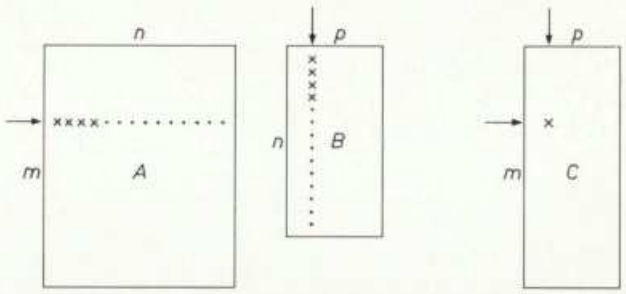


Fig. 4. Illustratie van de dimensies van de matrices in het produkt $C^p_m = A^n_m \, B^p_n$.

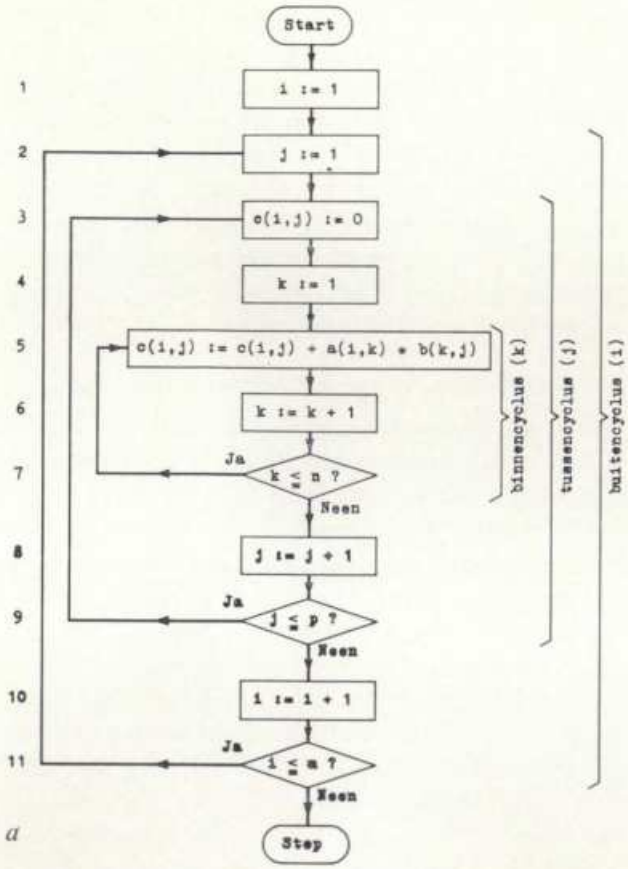
Een algoritme hiervoor is in *fig. 5a* weergegeven. Ter identificatie krijgt elke bewerking van het blokschema een volgnummer (links). De betekenis van de variabelen spreekt in dit geval voor zichzelf, maar volledigheidshalve geven wij in fig. 5*b* ook de tabel van symbolen, waarvan elk blokschema vergezeld behoort te gaan.

De kern van de algoritme is de operatie 5, die de in uitdrukking (6) voorkomende produkten bij elkaar optelt. Terwijl uitdrukking (6) alleen zegt wat het resultaat van de berekening moet zijn, geeft de algoritme ondubbelzinnig aan in welke volgorde de elementen van C berekend worden (rij voor rij) en in welke volgorde de termen worden opgeteld (elke rij van A wordt van links naar rechts doorlopen, en elke kolom van B van boven naar beneden).

Om te verduidelijken hoe de algoritme werkt en

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023



a

Symbool	Betekenis
i	rij-index van matrices A en C
j	kolom-index van matrices B en C
k	rij-index van matrix B / kolomindex van matrix A
m	aantal rijen in matrices A en C
n	aantal kolommen in matrix A / aantal rijen in matrix B
p	aantal kolommen in matrices B en C
$a(i,k)$	element van matrix A
$b(k,j)$	element van matrix B
$c(i,j)$	element van matrix C

b

Fig. 5. *a*) Fundamenteel blokschema van de algoritme voor het berekenen van het matrixprodukt $C = A \times B$. In FORTRAN wordt voor het vermenigvuldigen het teken * gebruikt om verwarring met de letter x te vermijden — een gewoonte die wij voor alle blokschema's hebben overgenomen.
b) Tabel van variabelen in het blokschema (*a*). Zulk een tabel vergemakkelijkt het lezen van het blokschema.

tevens de juistheid ervan na te gaan, passen we haar toe op de eenvoudige matrices

$$A = |1 \quad 2| \quad \text{en} \quad B = \begin{vmatrix} 1 & 3 \\ 2 & 4 \end{vmatrix}.$$

We hebben dus $m = 1$, $n = 2$, $p = 2$. De gang van zaken en de resultaten zijn in tabelvorm in *fig. 6* weergegeven. De eerste kolom bevat de nummers van de bewerkingen in volgorde van uitvoering. De volgende kolommen geven de variabelen van het blokschema. De laatste kolom geeft voor elke voorkomende vertakkingsopdracht aan of op de daarin gestelde vraag het antwoord „ja" of „neen" gegeven wordt. De eerste regel van de tabel stelt de beginsituatie voor: de variabelen hebben hier willekeurige, niet nader bekende waarden. Elke volgende regel wordt verkregen door in het blokschema na te gaan welke bewerking aan de beurt is, deze uit te voeren, en de resulterende waarde van elke variabele of het antwoord op een eventuele vraag in de bijbehorende kolom te schrijven. Bij elke stap van de algoritme (overeenkomende met een regel van de tabel) heeft een variabele de laatste in de desbetreffende kolom ingevulde waarde. De eindwaarden worden in de laatste regel (gemerkt „stop") samengevat en geven het resultaat weer:

$$C = |5 \quad 11|.$$

Aldus heeft men een vrijwel volledige controle op de algoritme. Voor systematische toepassing is deze boekhouding te omslachtig, maar ze is vaak van nut om

Label van de bewerking	Variabelen					Conditie voor de vertakking
	i	j	k	$c(1,1)$	$c(1,2)$	
start	—	—	—	—	—	—
1	1					
2		1				
3				0		
4			1			
5				1		
6			2			
7						ja
5				5		
6			3			
7						neen
8		2				
9						ja
3					0	
4			1			
5					3	
6			2			
7						ja
5					11	
6			3			
7						neen
8		3				
9						neen
10	2					
11						neen
stop	2	3	3	5	11	neen

Fig. 6. „Filmopname" van de uitvoering van de algoritme in fig. 5*a* voor de matrixvermenigvuldiging, met

$$A = |1 \quad 2|, \quad B = \begin{vmatrix} 1 & 3 \\ 2 & 4 \end{vmatrix}, \quad m = 1, \quad n = 2, \quad p = 2.$$

kritieke punten in het logische geheel na te gaan, zoals b.v. of de cycli het juiste aantal malen worden uitgevoerd, of de algoritme voor kritieke waarden van parameters kan worden toegepast, enz.

De voor de blokschema's ingevoerde notatie is natuurlijk door de structuur van de computers beïnvloed. Aan een dynamische variabele beantwoordt een ge-

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

heugenvakje. Aan de toekenning van een waarde aan zo'n variabele beantwoordt het opbergen van die waarde in het overeenkomstige geheugenvakje. Vertakkingsoperaties komen overeen met spronginstructies van de computer. Aan de volgnummers van bewerkingen beantwoorden adressen van instructies. De tabel in fig. 6 is niets anders dan een soort filmopname van de geheugeninhoud tijdens de uitvoering van het programma. Het blokschema kan derhalve worden gezien als een verkorte notatie voor een computerprogramma. Het stelt in feite de gedetailleerde specificatie voor aan de hand waarvan het programma geschreven wordt, zonder dat het, in de tot dusver beschouwde vorm, op een speciale computer is toegespitst. Deze laatste kan heel goed ook een menselijke rekenaar zijn, zoals uit de uitvoering van de algoritme voor het berekenen van het matrixprodukt blijkt. Deze algemeenheid van het blokschema maakt het bijzonder waardevol. Men kan er de gang van zaken uit aflezen zonder kennis van een speciale programmeertaal. Het geeft een maatstaf voor de beoordeling van de geschiktheid van een programma, en vooral ook een houvast om codeerfouten bij het programmeren en logische fouten in de opzet van elkaar te onderscheiden.

Het ontwerpen van een rekenschema is blijkens het bovenstaande een proces dat zich in een aantal etappes voltrekt. Het door de programmaspecificatie beschreven verband wordt ontleed in een reeks bewerkingen, waarvan de logische samenhang in een primair blokschema wordt weergegeven. Iedere niet-elementaire operatie vormt een deelprobleem, dat op dezelfde wijze behandeld wordt. Zo ontstaan secundaire blokschema's, en dit proces gaat door totdat het vraagstuk ten slotte is gereduceerd tot elementaire operaties. De definitie van „elementair" hangt overigens nog af van de computertaal en de beschikbare subroutines, zoals we hieronder zullen zien.

Slechts zelden zal men op deze wijze in één keer met het opzetten van het rekenschema klaarkomen. De op een bepaalde trap gekozen onderverdeling in deelproblemen kan ongeschikt blijken wanneer men deze nader uitwerkt. Sommige deelproblemen kunnen gelijkenis vertonen, hetgeen aanleiding kan geven tot een geheel nieuwe aanpak. Daarom is het opzetten van een rekenschema een iteratief proces. Het feit dat telkens een nieuwe aanpak mogelijk is nog zonder dat men zich om regels van een computertaal behoeft te bekommeren is juist een van de grote voordelen van de beschrijvingswijze met het blokschema.

Taak 3: het eigenlijke programmeren

De volgende stap bij het uitwerken van een programma bestaat in het uitschrijven van het opgestelde rekenschema in een taal die het computersysteem „begrijpt". Als we nu op het onderwerp van de computertalen willen ingaan, is het misschien het gemakkelijkst te beginnen met de hogere programmeertalen (voorschriftentalen; Engels: statement-type programming languages), hoewel deze historisch de jongste zijn. Deze omkering lijkt gerechtvaardigd, omdat de ontwikkeling van programmeertalen heel natuurlijk erop gericht is geweest het programmeren te vereenvoudigen, al werden daarvoor in het computersysteem als geheel ingewikkelder voorzieningen nodig.

Het programmeren in een hogere programmeertaal

Met een hogere programmeertaal kan men bewerkingen omschrijven op een hogere trap dan met de computerinstructies zelf. Deze talen zijn grotendeels onafhankelijk van de computer, en hun gebruik vereist dus geen kennis van een bepaalde machinecode, hoewel zulk een kennis wel tot een efficiëntere programmering kan bijdragen, zoals we hieronder zullen zien.

Om een programma dat geschreven is in een hogere programmeertaal op de computer te kunnen uitvoeren, moeten er middelen beschikbaar zijn om het in computerinstructies te vertalen. Deze vertaling geschiedt door „compileerprogramma's", welke het ontstaan van hogere programmeertalen duidelijk hebben beïnvloed. Zo is bij de eerste taal van deze soort, FORTRAN [5] (afkorting van „formula translation"), duidelijk rekening gehouden met het vertaalprobleem, terwijl recentere talen, zoals ALGOL 60 [6] (algorithmic language), een meer systematische opbouw hebben gekregen.

Ter illustratie geven we in *fig. 7* en *fig. 8* een gedeelte weer van een FORTRAN- en een ALGOL 60-programma waarmee het reeds besproken produkt van

```
1        DØ 10, I = 1(1)N
2        DØ 10, J = 1(1)P
3        C(I,J) = 0
4        DØ 10, K = 1(1)N
5   10   C(I,J) = C(I,J) + A(I,K) * B(K,J)
```

Fig. 7. Kern van een FORTRAN-programma voor het berekenen van het matrixprodukt $C = A \times B$. In FORTRAN-programma's wordt de letter O doorgestreept, om deze niet met het cijfer 0 te verwarren.

```
1  for i := 1 step 1 until m do
2  for j := 1 step 1 until p do
3      begin
4          c[i,j] := 0;
5          for k := 1 step 1 until n do
6          c[i,j] := c[i,j] + a[i,k] × b[k,j]
7      end
```

Fig. 8. Kern van een ALGOL 60-programma voor het berekenen van het matrixprodukt $C = A \times B$.

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

twee matrices kan worden berekend. Beide zijn gebaseerd op de algoritme volgens het blokschema fig. 5*a*. De voorschriften waarin de bewerkingen van het blokschema worden vertaald voldoen aan strenge taalregels. Zo wordt bewerking 5 van het schema (fig. 5*a*) uitgedrukt door het voorschrift 6 in het ALGOL 60-programma. De voorschriften worden door een puntkomma van elkaar gescheiden; deze heeft enigszins de functie van de begrenzing door het rechthoekige blokje in het blokschema. Het overeenkomstige FORTRAN-voorschrift 5 verschilt van het ALGOL 60-voorschrift door de notatie voor de operatoren en door het feit dat de namen van variabelen met hoofdletters moeten worden aangeduid. Het einde van een voorschrift wordt niet door een apart symbool aangegeven, maar blijkt uit het feit dat men op een nieuwe regel begint.

Een reeks handelingen van het blokschema wordt soms in een voorschriftentaal anders georganiseerd. Zo wordt het boekhouden voor de tussencyclus (j) (handelingen 2, 8 en 9 van het blokschema) in FORTRAN uitgevoerd door voorschrift 2 in combinatie met de label 10 van voorschrift 5. De betekenis ervan is dat de reeks voorschriften beginnende met 3 t/m 5 moet worden herhaald voor de gehele waarden 1 t/m P van de index J. In ALGOL 60 wordt dit uitgedrukt door het voorschrift 2, en het bereik hiervan wordt aangegeven door de combinatie **begin** (regel 3) en **end** (regel 7). Deze combinatie heeft een soortgelijke betekenis als haakjes in een rekenkundige uitdrukking.

Voor het programmeren van eenvoudige berekeningen zoals de hier behandelde vermenigvuldiging van matrices is eigenlijk een zo gedetailleerd blokschema als in fig. 5 niet nodig. In het algemeen echter zijn blokschema's, ook als men hogere programmeertalen gebruikt, nog wel degelijk nuttig. Ze blijven een goed middel om de samenhang van ingewikkelde rekenschema's tot uitdrukking te brengen.

Het programmeren in symbolische talen

Een hogere programmeertaal is niet altijd het aangewezen medium, en soms zelfs geheel ongeschikt. Met name geldt dit voor het schrijven van een compileerprogramma: dit moet geschreven zijn in een taal waarvoor reeds een vertaler aanwezig is. Een algemeen bezwaar van hogere programmeertalen is dat deze voor sommige soorten gegevens geen geschikte voorschriften bevatten om ze efficiënt onder te brengen en te verwerken. Als voordeel van de hogere programmeertalen hebben we genoemd dat zij onafhankelijk zijn van de computer, maar de keerzijde hiervan is dat met zulk een taal de mogelijkheden van het instructierepertoire van een bepaalde computer niet volledig kunnen worden benut. Dit kan wèl met talen die met behulp van een symbolisch formalisme de individuele computerinstructies uitdrukken. Zulk een taal, die geldt voor een bepaalde computer, heet een symbolische taal [7].

Voor het programmeren in een symbolische taal moet men een gedetailleerde kennis van de computerinstructies, de interne representatie van informatie, de indexregisters enz. bezitten. Er zijn ook meer gedetailleerde blokschema's nodig. In de praktijk wordt de symbolische code dan ook gewoonlijk nog slechts gebruikt door programmeurs binnen een rekencentrum, voor programma's waarbij hoge eisen aan de efficiëntie worden gesteld. Toch kan enig inzicht in het programmeren in een symbolische taal zijn nut hebben om een indruk te krijgen van de te verwachten verschillen tussen het resultaat van een compilatie uit een hogere programmeertaal en een rechtstreeks in symbolische taal geschreven programma.

Ter illustratie beschouwen we opnieuw de in fig. 5 uitgestippelde vermenigvuldiging van twee matrices. Over het algemeen bezitten computers instructies waardoor gewerkt kan worden met variabelen met één index (vectoren), maar niet met twee indices (matrices). Om deze beperking te omzeilen, stelt men matrices als vectoren voor, b.v. door de rij-vectoren van de matrix in volgorde achter elkaar te zetten. Zo wordt de hierboven gedefinieerde matrix C, die m rijen en p kolommen heeft, voorgesteld als een vector $\mathbf{C}_1$ met $m \times p$ componenten. De waarden van de vectorcomponenten $c1(r)$ corresponderen met de matrixelementen $c(i,j)$ door de betrekking tussen de indices:

$$\begin{aligned} & & r &= (i-1)\times p + j \\ & \text{voor} & i &= 1, 2, \ldots m \\ & & j &= 1, 2, \ldots p \\ & \text{en derhalve} & r &= 1, 2, \ldots m \times p. \end{aligned}$$

Overeenkomstig kunnen de matrices A en B worden voorgesteld door vectoren $\mathbf{A}_1$ en $\mathbf{B}_1$. De correspondentie is hier:

$$\begin{aligned} & & a1(s) &= a(i,k) \\ & \text{met} & s &= (i-1)\times n + k \\ & \text{voor} & i &= 1, 2, \ldots m \\ & & k &= 1, 2, \ldots n \\ & \text{en derhalve} & s &= 1, 2, \ldots m \times n, \end{aligned}$$

en evenzo:

$$\begin{aligned} & & b1(t) &= b(k,j) \\ & \text{met} & t &= (k-1)\times p + j \\ & \text{voor} & k &= 1, 2, \ldots n \\ & & j &= 1, 2, \ldots p \\ & \text{en derhalve} & t &= 1, 2, \ldots n \times p. \end{aligned}$$

[5] D. McCracken en W. Dorn, Numerical analysis and FORTRAN programming, Wiley, New York 1964.

[6] Revised report on the algorithmic language ALGOL 60, red. Peter Naur, A/S Regnecentralen, Kopenhagen 1964. Zie ook b.v. E. W. Dijkstra, A primer of ALGOL 60 programming, Academic Press, Londen 1962.

[7] P. Wegner, Introduction to symbolic programming, Griffin, Londen 1963.

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

Voert men deze vectornotatie en de bijbehorende indexbetrekkingen in het rekenschema van fig. 5 in, dan krijgt men het blokschema *fig. 9*. Er moet goed op gelet worden dat de bewerkingen voor de indexconversie de juiste plaats in het blokschema krijgen. Zo moet bewerking 3 voor de berekening van de index r:

$$r := (i-1)\times p + j$$

worden uitgevoerd na elke verandering van i of j en vóór elke bewerking waarin $c1(r)$ voorkomt.

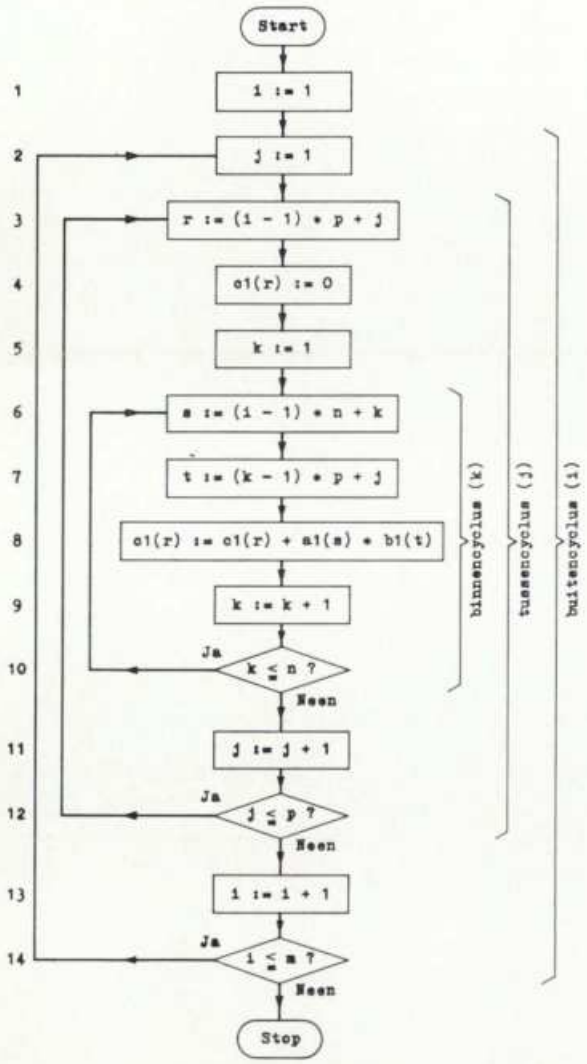


Fig. 9. Blokschema voor een matrixvermenigvuldiging, zodanig geschreven dat geen variabelen met meer dan één index voorkomen. Geschikt voor programmering in een symbolische taal.

Het blokschema fig. 9 bevat alle variabelen en bewerkingen die voor het programmeren in een symbolische taal nodig zijn. In de eindprogramma's (object programs), verkregen door vertaling (compilatie) van de in fig. 7 en 8 weergegeven FORTRAN- en ALGOL 60-programma's, zullen voor de indexconversie soortgelijke bewerkingen als 3, 6 en 7 optreden. Voor deze indexconversies zal een belangrijk gedeelte van de totale rekentijd nodig zijn. Dit ziet men in fig. 9, als men bedenkt dat de bewerkingen 6 t/m 10 van de binnencyclus (k) hier niet minder dan $m\times n\times p$ maal worden herhaald, d.i. een veelvoud van het aantal malen van de andere bewerkingen (behalve in het triviale geval dat $n = 1$). Tellen we voor een globale schatting de elementaire rekenoperaties in de binnencyclus, dan vinden we dat de bewerkingen met de matrixvariabelen slechts neerkomen op één optelling en één vermenigvuldiging, terwijl het opereren met de indices vijf optellingen en twee vermenigvuldigingen kost. Aangezien het vermenigvuldigen bij de meeste computers veel langer duurt dan het optellen, loont het de moeite, een strategie voor de indexberekening te zoeken waarbij géén vermenigvuldigingen voorkomen. Dit komt erop neer dat we de wetmatigheden in de opeenvolging van de vectorindices r, s en t opsporen, en dat we deze wetmatigheden rechtstreeks in formule brengen in plaats van het verband met de matrixindices i, j en k op te stellen. Het resultaat van deze herziening van de indexberekening ziet men in *fig. 10*. De details hiervan zijn voor ons betoog niet van belang, maar als we de binnencyclus (k) van fig. 10 bekijken, zien we dat het opereren met de indices nu slechts drie optellingen kost. Ten opzichte van fig. 9 worden dus twee vermenigvuldigingen uitgespaard. Dit brengt b.v. op de Philips P3 computer (een getransistoriseerde versie van de vroeger beschreven PASCAL [8]) een verkorting van de rekentijd van 40% met zich mee. Voor een programma dat slechts een enkele maal dienst moet doen is dit misschien niet van belang, maar het ligt voor de hand dat een programma als het hier bedoelde zal dienen als een subroutine, d.w.z. als een rekenprocédé dat als één geheel in allerlei andere programma's kan worden opgenomen. Het werk, besteed aan de omzetting van het blokschema fig. 9 in fig. 10, zal dus een zinvolle investering zijn.

Met nadruk zij erop gewezen dat het blokschema het juiste stadium is voor het analyseren en aanbrengen van zulke wijzigingen in het rekenschema. Overigens kost het programmeren van het blokschema van fig. 10 niet meer moeite dan het programmeren van het schema van fig. 9.

Hierboven hebben wij bekeken hoe blokschema's getransformeerd en verfijnd kunnen worden, om de meer gedetailleerde rekenschema's te verkrijgen die voor het schrijven in symbolische taal nodig zijn. We gaan nu

[8] W. Nijenhuis, De PASCAL, een zeer snelle digitale elektronische rekenmachine voor Philips' Rekencentrum, Philips techn. T. **23**, 1-19, 1961. Zie ook H. J. Heijn en J. C. Selman, The Philips computer PASCAL, IRE Trans. on Electronic Computers **EC-10**, 175-183, 1961.

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

dit programmeren zelf behandelen, en we doen dit kortheidshalve aan de hand van een nog eenvoudiger voorbeeld.

Stel we willen de veelterm

$$P(x) = a_0 x^n + a_1 x^{n-1} + \ldots + a_n$$

berekenen. Als we deze in de vorm

$$P(x) = \{\ldots [(a_0 x + a_1)x + a_2]x + \ldots a_{n-1}\}x + a_n$$

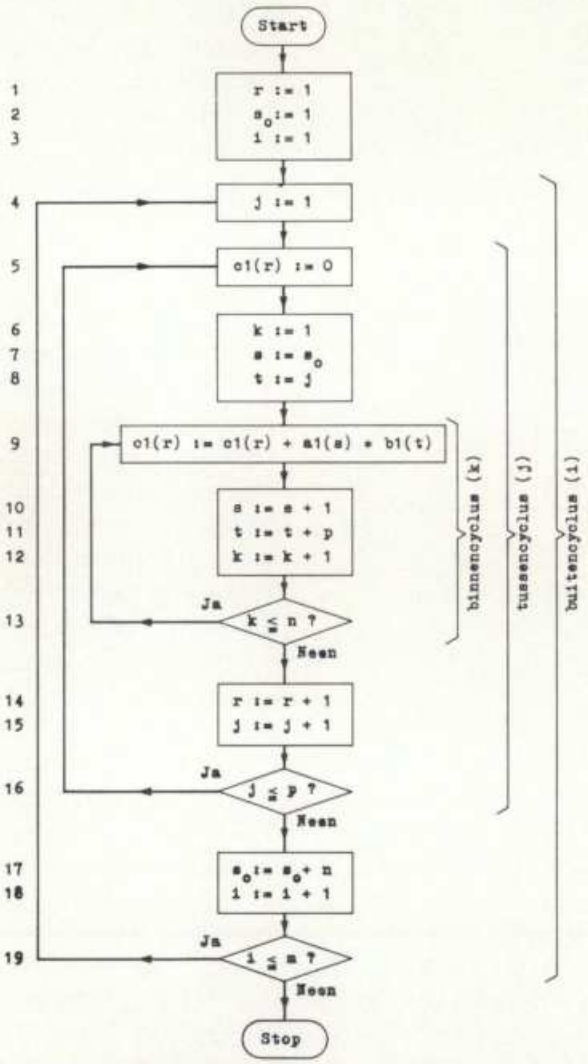
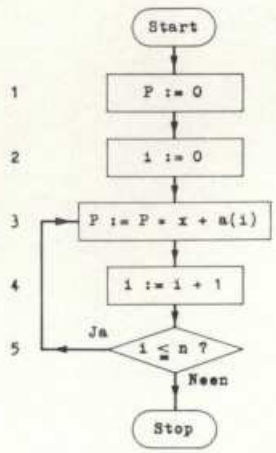


Fig. 10. Efficiënter blokschema voor de matrixvermenigvuldiging. De indices i en j van de matrix C worden hierbij in zodanige volgorde doorlopen dat de elementen van de vector $\mathbf{C}_1$ in natuurlijke volgorde van de index r worden verkregen. Door bewerking 1 wordt r op 1 gesteld, en door bewerking 14 wordt telkens r met 1 verhoogd na beëindiging van de binnencyclus (k). Om de kolom j van de matrix B af te tasten, moet de index t van de vector $\mathbf{B}_1$ met de waarde j beginnen en voor elke stap in k met p worden vergroot. Vandaar de operaties 8 en 11 (dit proces is onafhankelijk van de rij-index i). De regelmaat in de toeneming van de index s van vector $\mathbf{A}_1$ is ingewikkelder. De matrix A wordt rij voor rij afgetast, zodat s met 1 toeneemt tegelijk met k (operatie 10). Maar bij elke j-cyclus houdt men dezelfde rij in A, zodat hier s telkens tot een beginwaarde s_0 wordt teruggebracht (operatie 7). Dit is de index van $\mathbf{A}_1$ die behoort bij het eerste element van de rij i van matrix A. Derhalve begint s_0 met de waarde 1 (operatie 2) en neemt met n toe voor elke stap in i (operatie 17).

schrijven, kan de berekening worden uitgevoerd volgens de algoritme die in het blokschema *fig. 11* is voorgesteld. We zullen nu bespreken hoe deze algoritme wordt geprogrammeerd.



Symbool van het blokschema	Symbool van het programma	Betekenis
P	PØL	waarde van de veelterm
x	X	argument
n	N	graad van de veelterm
a	AV	coëfficiëntenvector
i	IR1	index van de component

Fig. 11. Blokschema voor de berekening van een veelterm. Onderaan: de tabel van symbolen voor het programmeren in een symbolische taal.

Bij het programmeren moet alleereerst voor elk der variabelen in het blokschema een naam worden gekozen die voldoet aan de taalregels. Verder moeten indexregisters van de computer worden toegewezen aan de indices die in het blokschema voorkomen (zijn er niet voldoende beschikbaar, dan moet men instructies toevoegen, die de indices *om beurten* opbergen). Al deze toewijzingen vat men in een tabel samen, die later de leesbaarheid van het programma zeer verhoogt, vooral als de taalregels geen naamgeving toestaan waaruit vanzelf volgt hoe een en ander aan elkaar toegevoegd is.

Bij het programmeren in een symbolische taal gaat men nu zo te werk dat iedere bewerking uit het schema in een reeks instructies wordt ontleed. Dit ziet men in de tabel van *fig. 12a*. In kolom 2 vindt men de operaties uit het instructierepertoire van de machine die nodig zijn, nog geschreven in de notatie van het blokschema, en in de kolommen 3-5 het symbolische programma. Elke regel heeft gemakshalve een volgnummer (kolom 1). De namen A en S in kolom 2 duiden rekenregisters aan die bij de ingebouwde (in microprogramma's vastge-

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

a

Volg-nummer	Bewerking	Instructie: label	Instructie: operatie-deel	Instructie: operand	Adres in geheugen
1	$A := 0$		HIA	NUL	512
2	$P := A$		BAP	PØL	513
3	$i := A$		BAP	IR1	514
4	$S := x$	CYCLUS	HIS	X	515
5	$A := x \times P$		VMG	PØL	516
6	$A := A + a(i)$		ØPA	AV,1	517
7	$P := A$		BAP	PØL	518
8	$A := i$		HIA	IR1	519
9	$A := A + 1$		ØPA	EEN	520
10	$i := A$		BAP	IR1	521
11	$A := A - n$		AFA	N	522
12	naar 4 als $A \leqq 0$		SAN	CYCLUS	523
13	stop		STP		524
14		NUL	FLN	0	525
15		EEN	FLN	1	526
16		PØL	ØPN		527
17		X	ØPN		528
18		N	ØPN		529
19		AV	ØPN	50	530

b

Operatiecode: numeriek	Operatiecode: symbolisch	Omschrijving van operatie
00	HIA	Haal operand in *A*-register
01	HIS	Haal operand in *S*-register
02	BAP	Berg inhoud van *A*-register weg op adres van de operand
03	ØPA	Tel operand op bij *A*-register
04	AFA	Trek operand af van *A*-register
05	VMG	Vermenigvuldig operand met *S*-register
06	SAN	Spring naar instructielabel gegeven door operand, indien *A* negatief of nul is
07	STP	Stop

Fig. 12. *a*) Programma in symbolische taal voor de berekening van een veelterm. In kolom 2 zijn de bewerkingen in de notatie van het blokschema aangegeven. In de kolommen 3-5 vindt men de symbolische code voor het programma. *b*) Betekenis van het operatiedeel in de symbolische code van (*a*).

legde) elementaire rekenoperaties gebruikt worden. Het symbolische programma bestaat uit de rij instructies 1 t/m 13, gevolgd door de opsomming in regel 14 t/m 19 van de waarden van de variabelen. Elke instructie bestaat uit drie gedeelten: een label ten behoeve van verwijzingen (indien nodig), een operatiedeel (zie fig. 12*b*) en een operand, welke elders optreedt als label voor een variabele of een instructie. De specificaties van waarden voor de variabelen (regel 14 t/m 19) hebben een soortgelijke bouw; deze zal besproken worden in samenhang met het vertalen van de symbolische taal.

Op een paar punten zullen we nu de samenhang tussen het blokschema en het programma nagaan.

Bewerking 3 van het blokschema:

$$P := P \times x + a(i),$$

wordt weergegeven door de vier instructies 4 t/m 7. Instructie 4 brengt de waarde van x naar het *S*-register. Instructie 5 neemt de *S*-waarde als vermenigvuldiger en *P* als vermenigvuldigtal en levert het produkt af in het *A*-register. De operand AV,1 van instructie 6 (de 1 achter de komma verwijst naar een indexregister) is die component $a(i)$ van de vector **AV** waarvan het rangnummer door de inhoud i van het indexregister *IR*1 wordt aangegeven. Instructie 6 telt dus deze component op bij het produkt $P \times x$ dat in *A* aanwezig is, en instructie 7 bergt het resultaat op in *P*.

Bewerking 5 van het blokschema wordt gerealiseerd door de reeks instructies 10 t/m 12, waarbij 12 een sprong is die afhangt van het teken van *A*. Deze reeks is bruikbaar indien een uitkomst 0 van de instructie 11, voor $i = n$, in de computer wordt voorgesteld met een negatief teken, hetgeen bij vele computers het geval is.

Programmeren in symbolische taal vergt, zoals hieruit blijkt, zeer veel aandacht voor allerlei kleinigheden. Daarom is het van groot belang dat men bij het schrijven van het programma kan bouwen op een goede opzet van het rekenschema en zich niet meer om de logica van deze opzet behoeft te bekommeren. Zelfs dan moet men niet verwachten dat een groot programma al in eerste ontwerp goed zal zijn. Het is derhalve verstandig reeds van tevoren met fouten rekening te houden en maatregelen te nemen om deze te kunnen opsporen en verbeteren. Dit is het zg. „debugging". Methodes hiervoor zijn: onderverdeling van programma's (segmentatie), ontwikkeling van speciale testprogramma's, zichtbaar maken van tussenresultaten, enz.

Ook hier geldt dat de taken niet scherp begrensd zijn. Voor een efficiënter programma kunnen wijzigingen van de opzet, of althans van details ervan, gewenst zijn. De efficiëntie betreft het gehele proces van schrijven en debugging, maar men verwarre dit niet met de toepassing van kunstgrepen om een programma zo kort mogelijk op te schrijven. Een bekende kunstgreep bestaat daarin dat tijdens de uitvoering van een programma een reeks instructies wordt gewijzigd door de instructies zelf als operanden te gebruiken — een aardigheid die uit de pionierstijd, toen computers nog geen indexregisters hadden, overgebleven is en waar-

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

mee sommige programmeurs zich nog vermaken, die er dan ook in slagen hun programma's voor anderen vrijwel onleesbaar te maken.

De machinetaal

Voor de machinetaal geldt a fortiori wat hierboven gezegd is voor symbolische talen: ofschoon de machinetaal door bijna niemand meer zal worden gebruikt, lijkt het ons zinvol er enige aandacht aan te besteden ten behoeve van de vertaling. Het programma in machinetaal is immers van elke vertaling het uiteindelijk resultaat — het is de enige taal die de computer in feite verstaat.

Een programma in symbolische taal is een getrouw beeld van de gecodeerde informatie die tijdens de uitvoering in het geheugen van de computer zal staan. Het plaatsen in het geheugen — d.i. in feite de vertaling in machinecode — gebeurt automatisch door een apart, zelf natuurlijk in machinecode geschreven, programma („assembler"-programma). Hierbij wordt een lijst gemaakt die geheugenadressen in de machine toekent aan labels van instructies en variabelen. Met behulp van deze lijst worden in de operandgedeelten van de instructies de symbolische adressen vervangen door numerieke. Het invullen van waarden voor de variabelen die als operanden voorkomen wordt teweeggebracht door middel van speciale opdrachten in de tekst van het symbolische programma, die alleen een aanwijzing zijn voor de vertaler maar geen instructies in het eigenlijke machineprogramma tot gevolg hebben. Zo stellen de regels 14 t/m 19 in fig. 12 geen eigenlijke machine-instructies voor, maar opdrachten die geheugencellen toewijzen aan de variabelen van het programma en die deze cellen hetzij vullen met de waarden die de betrokken variabelen in het programma hebben moeten, zoals de constanten „nul" en „een" (in zwevendekomma-notatie), hetzij deze waarden nog open laten. Zo reserveert regel 19 de 50 geheugenplaatsen van 530 t/m 579 ten behoeve van de coëfficiënten $a(i)$.

Ter illustratie geeft *fig. 13* de lijst van machine-instructies voor het berekenen van een veelterm in een hypothetische machinecode. Dit programma kan uit het symbolische programma fig. 12 worden afgeleid door na te doen wat anders bij de automatische machinevertaling gebeurt. Allereerst kent men opeenvolgende adressen toe aan de instructies, zoals in de laatste kolom van fig. 12*a* is aangegeven. Het beginadres (512) is willekeurig, maar de volgorde beantwoordt aan de wijze waarop de computerbesturing werkt, die ervoor zorgt dat in de tijd opeenvolgende instructies uit opeenvolgende geheugenplaatsen worden gehaald en uitgevoerd, tenzij de instructie zelf iets anders voorschrijft (sprongen). De zo verkregen wederkerige toevoeging van labels en numerieke adressen wordt samengevat in een dictionaire (*fig. 14*). Hiermee worden de symbolische operanden in het programma fig. 12*a* vervangen. Als men daarna nog de symbolisch aangegeven operatoren door hun numerieke codering vervangt (fig. 12*b*), krijgt men het programma van fig. 13, dat in numerieke code de inhoud van elke betrokken geheugencel aangeeft. Afgezien van een eventuele omzetting in binaire vorm, is dit de werkelijke gedaante waarin het programma tijdens de uitvoering in het geheugen staat. Om het daarin te brengen, heeft men nog slechts een zeer eenvoudig invoerprogramma nodig voor het lezen van de kaarten of de papierband waarin de tekst van fig. 13 is geponst. Dit eenvoudige invoerprogramma is meestal in de vorm van een bedrading vast in de machine aanwezig.

Adres	Operatie-code	Index-register	Operand-adres	
512	00	0	0525	
513	02	0	0527	
514	02	0	0032	
515	01	0	0528	
516	05	0	0527	
517	03	1	0530	
518	02	0	0527	
519	00	0	0032	
520	03	0	0526	
521	02	0	0032	
522	04	0	0529	
523	06	0	0515	
524	07	0	0000	
525	. .	.	. . . .	= „0"
526	. .	.	. . . .	= „1"
527	. .	.	. . . .	= „*P*"
528	. .	.	. . . .	= „*x*"
529	. .	.	. . . .	= „*n*"
530	. .	.	. . . .	= „*a*(1)"
.				.
.				.
.				.

Fig. 13. Het programma voor de berekening van een veelterm in een hypothetische machinecode. Het programma is de vertaling van het symbolische programma van fig. 12*a*. In kolom 3 is aangegeven of een, en zo ja, welk indexregister gebruikt moet worden ter modificatie van het adresgedeelte; waar 0 is ingevuld, heeft geen modificatie plaats. De adressen 525 enz. bevatten de numerieke waarden van de variabelen die in aanhalingstekens vermeld zijn.

Label	Adres
AV	530
CYCLUS	515
EEN	526
IR1	32
N	529
NUL	525
PØL	527
X	528

Fig. 14. Lijst van geheugenadressen die corresponderen met de labels van het programma fig. 12*a*.

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

Programmeertalen en vertalers

Aan het eind van dit korte overzicht van het programmeren in verschillende talen is het interessant, nogeens terug te zien en de programmeertalen in hun historische ontwikkeling te beschouwen [9]. Het moeizame werk van het programmeren in machinetaal gaf al vroeg aanleiding tot de ontwikkeling van symbolische talen. Een symbolische taal heeft een equivalent voor iedere machine-instructie en is dus bijna even flexibel als de machinetaal zelf. Derhalve is het coderen in machinetaal, waarbij men expliciet geheugenadressen en code-aanduidingen van instructies moet geven, praktisch verdwenen, behalve misschien voor de zeer fundamentele programma's en subroutines ten behoeve van de interne organisatie binnen een computersysteem.

Toen men ging proberen computer-onafhankelijke programma's te schrijven, kwam men tot de programmeertalen van het voorschriftentype. Deze ontstonden doordat men zich ging realiseren dat de vertaling van een voorschrift zoals

$$z := x + y$$

in een reeks instructies:

$$A := x$$
$$A := A + y$$
$$z := A$$

zelf een bewerking is waarvoor men een algoritme kan aangeven, die men door computerinstructies kan laten uitvoeren.

De oorspronkelijke bedoeling van FORTRAN was, het schrijven van rekenkundige uitdrukkingen met haakjes mogelijk te maken. Het hoofdprobleem daarbij was het ontwerpen van de vertaler die automatisch de vertaling in de uiteindelijke reeks machine-instructies, het „object program", volbrengt. De incidentele of principiële gebreken van FORTRAN, alsmede de ervaring opgedaan met vertaaltechnieken, leidden ertoe dat men zich speciaal ging interesseren voor de structuur en de beschrijving van programmeertalen. Een van de resultaten hiervan is ALGOL 60, en nieuwere versies zijn nog in wording. Tevens kan nu, dank zij de vooruitgang in syntactische analyse, het schrijven van de vertaalprogramma's gedeeltelijk worden geautomatiseerd [9].

De mogelijkheid van automatische vertaling ontslaat de programmeur van de verplichting de machinecode in detail te kennen. Natuurlijk moet men bij het gebruik van een hogere programmeertaal rekening houden met een langere rekentijd van de computer, niet alleen omdat het vertalen zelf tijd opeist, maar ook omdat het uiteindelijk resulterende machinecodeprogramma nauwelijks zo efficiënt zal zijn als een goed geschreven symbolisch programma. Dit verlies in efficiëntie is echter in het algemeen te verwaarlozen naast de tijdwinst die men heeft bij het schrijven van het programma en het elimineren van fouten. Toch is het nuttig zich ervan bewust te zijn hoe het resultaat van de vertaling van sommige voorschriften uit de hogere programmeertaal in de machinetaal uitvalt.

Neem b.v. nog eens het ALGOL 60-voorschrift 6 in fig. 8:

$$c[i,j] := c[i,j] + a[i,k] \times b[k,j].$$

Het uiteindelijke programma dat men hieruit door vertaling zou kunnen krijgen wordt weergegeven door de volgende reeks bewerkingen:

1	$r := (i-1) \times n + k$
2	$y := a1(r)$
3	$r := (k-1) \times p + j$
4	$y := b1(r) \times y$
5	$r := (i-1) \times p + j$
6	$y := y + c1(r)$
7	$r := (i-1) \times p + j$
8	$c1(r) := y.$

De variabelen hierin zijn als volgt gedefinieerd:

r	index van de vector die in de computer de representatie van de matrix is,
$a1$, $b1$, $c1$	vectorcomponenten die overeenkomen met de matrixcomponenten a, b, c als bovenbeschreven,
y	tussenvariabele.

De indexconversie

$$r := (i-1) \times p + j,$$

die het verband tussen de matrix C en zijn vectorrepresentatie $\mathbf{C}_1$ vastlegt, wordt hier twee keer herhaald, omdat er niet van uitgegaan wordt dat de vertaler het ALGOL 60-voorschrift zodanig onderzoekt dat hij ontdekt dat dezelfde component $c(i,j)$ twee keer voorkomt. De herhaling gebeurt zelfs liefst $m \times 2n \times p$ keer in het gehele programma, aangezien het voorschrift deel uitmaakt van de binnencyclus (k) van het rekenschema (fig. 5*a*), waarbinnen de index niet verandert.

Als op zulke bijzonderheden wordt gelet, kan de efficiëntie van een programma met weinig moeite aanzienlijk worden vergroot. Zo kan een kleine wijziging van het blokschema (fig. 5*a*) en van het ALGOL 60-programma (fig. 8) in het gegeven voorbeeld een aanzienlijke winst in rekentijd opleveren (bijna 50% met het Electrologica X8-ALGOL 60-systeem). De wijziging bestaat daarin dat men een hulpvariabele x invoert voor de som van alle reeds berekende deelprodukten in de k-cyclus (*fig. 15* en *16*). Het programma wordt

[9] J. Loeckx en M. Sintzoff, Les langages de programmation et leur traitement automatique, Rev. MBLE **9**, 126-146, 1966.

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

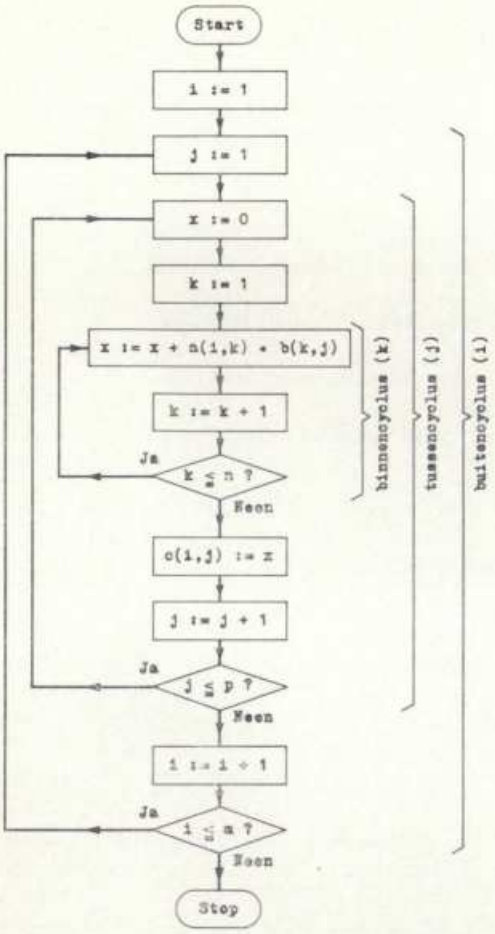


Fig. 15. Verbeterd blokschema voor de berekening van een matrixprodukt, ten behoeve van een hogere programmeertaal.

```
for i := 1 step 1 until m do
for j := 1 step 1 until p do
    begin
      x := 0;
      for k := 1 step 1 until n do
        x := x + a[i,k] × b[k,j];
      c[i,j] := x
    end
```

Fig. 16. Kern van verbeterd ALGOL 60-programma voor de berekening van het matrixprodukt $C = A \times B$.

hierdoor haast niet verlengd: er komt alleen aan de uitgang van de k-cyclus de opdracht bij die aan $c(i, j)$ de waarde x toekent.

Conclusies

Het opstellen van rekenmachineprogramma's voor wetenschappelijke problemen is een tijdrovend proces, waarbij de voornaamste moeilijkheden niet altijd daar liggen waar dit gewoonlijk wordt aangenomen. De diensten die de rekencentra bieden en het feit dat men beschikt over hogere programmeertalen die onafhankelijk zijn van een bepaalde computer hebben de computers gemakkelijker toegankelijk gemaakt. Het scherp stellen van het vraagstuk en het vele denkwerk bij het uitwerken van de oplossing blijven echter een aanzienlijke inspanning vergen. Dit moge misschien triviaal klinken, maar toch bestaat sinds de opkomst van de computer de haast onuitroeibare mythe dat een rekencentrum een soort universele probleemoplosinstantie is, waaraan men vaag geformuleerde problemen kan voorleggen om dan met enorme snelheid de oplossingen in ontvangst te nemen. De werkelijkheid lijkt eerder op het tegendeel. Weliswaar bevat de computer de middelen om instructies snel en betrouwbaar uit te voeren, maar voordat men een programma werkend heeft, moet men een lange en moeilijke weg gaan. Wij hebben hier getracht enkele facetten van dit voorbereidende werk te belichten, maar moesten diverse andere in het duister laten.

We hebben het voorbereidende werk in een reeks opeenvolgende taken onderverdeeld — niet alleen gemakshalve, maar omdat de verdeling beantwoordt aan het bereiken van bepaalde resultaten, als het ware mijlpalen, waarbij de verantwoording aan een volgende instantie overgedragen kan worden. De steller van het probleem zorgt voor de omschrijving ervan, de analist legt het rekenschema vast, de programmeur zorgt ervoor dat een correct programma geschreven wordt. Hoe men de taken en de uitvoerenden noemt, doet hierbij niet ter zake. Zo kan een programmeur verantwoordelijkheid dragen zowel voor de analyse als voor het schrijven van het programma. Het meest kritieke punt echter is het opstellen van de probleemomschrijving. Elke hierin nog overgebleven onnauwkeurigheid weerspiegelt zich in (eventueel onbewuste) afwijkingen in het programma, en tast de zekerheid aan dat de uitkomst het gewenste antwoord zal voorstellen.

Samenvatting. In het artikel wordt het proces van ontwerpen van computerprogramma's voor wetenschappelijke problemen geanalyseerd — dit vooral ten gerieve van onderzoekers en ingenieurs, die in de rekencentra steeds machtiger hulpmiddelen ter beschikking krijgen maar het gebruik hiervan steeds meer voor hun eigen rekening moeten nemen. Er wordt gepleit voor een methodische aanpak, met een verdeling in drie opeenvolgende taken: de systeemanalyse, het opzetten van het rekenschema en het eigenlijke programmeren. Deze verdeling bevordert dat bij elke stap de nodige strengheid in acht wordt genomen. Het programmeren door de gebruiker zelf geschiedt meestal in een „hogere programmeertaal". Tot slot worden ook symbolische en machine-talen in het kort besproken, om duidelijk te maken wat de functie van een vertaler is en wat er met een programma gebeurt als dit door de computer wordt uitgevoerd.

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

Recente wetenschappelijke publikaties

Deze publikaties zijn afkomstig van medewerkers van laboratoria en fabrieken, behorende tot of samenwerkende met ondernemingen van het Philipsconcern, in het bijzonder van medewerkers van de navolgende research-laboratoria:

Philips' Natuurkundig Laboratorium, Eindhoven	*E*
Mullard Research Laboratories, Redhill (Surrey), Engeland	*M*
Laboratoires d'Electronique et de Physique Appliquée, Limeil-Brévannes (Val-de-Marne), Frankrijk	*L*
Philips Zentrallaboratorium GmbH, Laboratorium Aken, Weisshausstrasse, 51 Aachen, Duitsland	*A*
Philips Zentrallaboratorium GmbH, Laboratorium Hamburg, Vogt-Kölln-Strasse 30, 2 Hamburg-Stellingen, Duitsland	*H*
MBLE Laboratoire de Recherches, 2 avenue Van Becelaere, Brussel 17 (Boitsfort), België.	*B*

Van de meeste dezer publikaties zullen binnenkort overdrukken beschikbaar zijn. Men kan deze aanvragen bij het betrokken laboratorium (zie de codeletter) of bij de Administratie van Philips' Natuurkundig Laboratorium, Eindhoven.

G. A. Acket: Galvanomagnetic effects of hot electrons in gallium arsenide.
Philips Res. Repts. **23**, 317-331, 1968 (No. 4). *E*

H. Bacchi & M. Blanchet (Services Techniques des Armées): Générateurs d'impulsions haute tension de 1 nanoseconde à mi-hauteur.
Onde électr. **48**, 430-437, 1968 (No. 494). *L*

P. Beekenkamp: On the structure of glasses in the system K_2O-B_2O_3-Al_2O_3.
Phys. Chem. Glasses **9**, 14-20, 1968 (No. 1). *E*

P. Beekenkamp & J. M. Stevels: A suggestion for a new nomenclature to describe the structure of vitreous systems and their related crystalline compounds.
Phys. Chem. Glasses **9**, 64-68, 1968 (No. 2). *E*

W. J. H. Beekman: On a new type of X-ray generator.
Thesis, Eindhoven 1967. *E*

G. Blasse: Some considerations and experiments on concentration quenching of characteristic broad-band fluorescence.
Philips Res. Repts. **23**, 344-361, 1968 (No. 4). *E*

G. Blasse, W. L. Wanmaker (Philips Lighting Division, Eindhoven) **& J. W. ter Vrugt** (Philips L.D.): Some new classes of efficient Eu^{2+}-activated phosphors.
J. Electrochem. Soc. **115**, 673, 1968 (No. 6). *E*

P. F. Bongers & G. Zanmarchi: Infrared absorption spectrum and Faraday rotation of ferromagnetic $CdCr_2Se_4$.
Solid State Comm. **6**, 291-294, 1968 (No. 5). *E*

A. H. Boonstra: Some investigations on germanium and silicon surfaces.
Thesis, Eindhoven 1967. *E*

G. Bosch: A Hall device in an integrated circuit.
Solid-State Electronics **11**, 712-714, 1968 (No. 7). *E*

S. E. Bradshaw (Associated Semiconductor Manufacturers Ltd., Wembley, England): The effects of gas pressure and velocity on epitaxial silicon deposition by the hydrogen reduction of chlorosilanes.
Int. J. Electronics **23**, 381-391, 1967 (No. 4).

A. Bril, G. Blasse & J. A. A. Bertens: Measurement of quantum efficiencies of Eu^{3+}-activated phosphors using excitation to selected Eu^{3+}-levels.
J. Electrochem. Soc. **115**, 395-397, 1968 (No. 4). *E*

S. D. Brotherton & J. B. Phillips (Associated Semiconductor Manufacturers Ltd., Wembley, England): The control of threshold voltage of complementary pair m.o.s. transistors.
IEE Conf. Publn. 30: Integrated circuits, p. 126-131, 1967.

K. H. J. Buschow: Crystalline Stark splitting in $TmAl_3$.
Z. phys. Chemie Neue Folge **59**, 21-26, 1968 (No. 1-4). *E*

H. B. G. Casimir: Betrachtungen über integrierte Schaltungen und Mikrotechnik.
Tagungsbroschüre VDE-Fachtagung Elektronik 1968, 6 pp. *E*

T. Chisholm: Influence of space charge on the Langmuir limit for focused electron beams.
IEEE Trans. **ED-15**, 374-381, 1968 (No. 6). *M*

C. Crevecoeur & H. J. de Wit: Thermoelectric force versus electrical resistivity of MnO at 1100 °C. An example of "Jonker's pear".
Solid State Comm. **6**, 295-296, 1968 (No. 5). *E*

A. Dekker & A. Middelhoek (Philips Electronic Components and Materials Division, Development Dept. Electrolytic Capacitors, Zwolle): The anodic oxidation of aluminium.
Philips Res. Repts. **23**, 342-343, 1968 (No. 4).

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

J. Drenth (Rijksuniversiteit Groningen), **W. G. J. Hol** (R.U. Gr.), **J. W. E. Visser** (R.U. Gr.) **& L. A. Æ. Sluyterman:** Papain in water-rich and in methanol-rich media; crystallization and conformation.
J. mol. Biol. **34**, 369-371, 1968 (No. 2). *E*

W. F. Druyvesteyn: Size-effects in the magnetoresistance of Al and In films.
Phil. Mag. **18**, 11-26, 1968 (No. 151). *E*

W. F. Druyvesteyn, A. J. Smets & C. A. A. J. Greebe: Experiments on Alfvén-wave resonances in liquid sodium.
Philips Res. Repts. **23**, 332-341, 1968 (No. 4). *E*

G. Engelsma: The influence of light of different spectral regions on the synthesis of phenolic compounds in gherkin seedlings, in relation to photomorphogenesis, IV. Mechanism of far-red action.
Acta bot. neerl. **17**, 85-89, 1968 (No. 2). *E*

U. Enz & H. van der Heide: Two new manifestations of the photomagnetic effect.
Solid State Comm. **6**, 347-349, 1968 (No. 6). *E*

G. Eschard & R. Polaert: Tube obturateur pour photographie ultra-rapide.
Onde électr. **48**, 426-429, 1968 (No. 494). *L*

B. Frank & J. Liebertz: Glasbildung in den Systemen CaO-Al_2O_3, SrO-Al_2O_3 und BaO-Al_2O_3 durch Abschrecken schmelzflüssiger Tröpfchen.
Glastechn. Ber. **41**, 253-254, 1968 (No. 6). *A*

J.-M. Goethals & P. Delsarte: On a class of majority-logic decodable cyclic codes.
IEEE Trans. **IT-14**, 182-188, 1968 (No. 2). *B*

G. Groh: Multiple imaging by means of point holograms.
Appl. Optics **7**, 1643-1644, 1968 (No. 8). *H*

C. Haas: Spin-disorder scattering and magnetoresistance of magnetic semiconductors.
Phys. Rev. **168**, 531-538, 1968 (No. 2). *E*

S. H. Hagen: Surface-barrier diodes on silicon carbide.
J. appl. Phys. **39**, 1458-1461, 1968 (No. 3). *E*

J. Haisma & K. Roozendaal (Philips Industrial Equipment Division, Eindhoven): Investigation of the behaviour of an expansion ejector in the low temperature region beyond the λ-transition of helium.
Proc. XIIth Int. Congress of Refrigeration, Madrid 1967, paper No. 1.37, 9 pp.; 1968. *E*

J. A. Kerr (Associated Semiconductor Manufacturers Ltd., Wembley, England) **& L. N. Large** (SERL, Baldock, England): Semiconductor devices made by ion implantation.
Proc. int. Conf. on applications of ion beams to semiconductor technology, Grenoble 1967, p. 601-618.

J. E. Knowles: The simulation of domain wall motion in square loop ferrites, and other aspects of wall behaviour.
Brit. J. appl. Phys. (J. Physics D), ser. 2, **1**, 821-830, 1968 (No. 7). *M*

E. Laviron (Commissariat à l'Energie Atomique), **C. Delmare** (Comm. En. At.) **& H. Bacchi:** Caméras électroniques de 1 ns de durée d'ouverture.
Onde électr. **48**, 421-425, 1968 (No. 494). *L*

W. Lems & Th. Holtwijk: Magnetization reversal of electrodeposited FeNi films.
J. Physique **29**, Colloque C 2, 140-143, 1968. *E*

J. Loeckx & M. Sintzoff: Génération mécanique de traducteurs pour langages formels.
Actes 5ème Congrès AFIRO, Assoc. Franç. Inform. Paris, 1967, p. 145-148. *B*

G. Meijer: Rapid growth inhibition of gherkin hypocotyls in blue light.
Acta bot. neerl. **17**, 9-14, 1968 (No. 1). *E*

K. Mouthaan & W. Fulop (Brunel University, London, Acton, England): On diffusion of electrons across a magnetic field.
J. appl. Phys. **39**, 2021-2023, 1968 (No. 4). *E*

J. M. van Nieuwland, H. L. Hagedoorn, N. Hazewindus & P. Kramer: Magnetic analogue of electric field configurations applied to the central region of an A.V.F. cyclotron.
Rev. sci. Instr. **39**, 1054-1055, 1968 (No. 7). *E*

W. C. Nieuwpoort & G. Blasse: Remark on the calculation of cohesion energies using the ionic model.
J. inorg. nucl. Chem. **30**, 1635-1637, 1968 (No. 6). *E*

H. Ocken & J. H. N. van Vucht: Phase equilibria and superconductivity in the molybdenum-platinum system.
J. less-common Met. **15**, 193-199, 1968 (No. 2). *E*

P. Penning & A. H. Goemans: Contrast in X-ray topographs of thin crystals.
Phil. Mag. **18**, 297-300, 1968 (No. 152). *E*

L. G. Pittaway, J. Smith, E. D. Fletcher & B. W. Nicholls: Laser-induced electron emission from tungsten, tantalum and lanthanum hexaboride using ruby and argon lasers.
Brit. J. appl. Phys. (J. Physics D), ser. 2, **1**, 711-719, 1968 (No. 6). *M*

J. Polman: Resonanties in een inhomogene plasmakolom.
Ned. T. Natuurk. **34**, 102-111, 1968 (No. 4). *E*

A. Rabenau & H. Rau: Hydrothermale Züchtung von Goldkristallen.
Naturwiss. **55**, 336-339, 1968 (No. 7). *A*

Digitized in Heiloo the Netherlands Not for sale!

www.freeservicemanuals.info 11/14/2023

E. Schwartz: Broadband matching of resonant circuits and circulators.
IEEE Trans. **MTT-16**, 158-165, 1968 (No. 3). *A*

J. G. Siekman & R. E. Morijn: The mechanism of welding with a sealed-off continuous CO_2-gas laser.
Philips Res. Repts. **23**, 367-374, 1968 (No. 4). *E*

J. G. Siekman & R. E. Morijn: A simple power-output meter specially designed for continuous laser beams.
Philips Res. Repts. **23**, 375-387, 1968 (No. 4). *E*

W. A. J. J. Velge & K. H. J. Buschow: Magnetic and crystallographic properties of some rare earth cobalt compounds with $CaZn_5$ structure.
J. appl. Phys. **39**, 1717-1720, 1968 (No. 3). *E*

A. G. van Vijfeijken: On the theory of vortices in type-II superconductors.
Thesis, Amsterdam 1967. *E*

W. L. Wanmaker & D. Radielović (Philips Lighting Division, Eindhoven): Luminescence of phosphates.
Bull. Soc. Chim. France 1968, 1785-1791 (No. spécial).

W. L. Wanmaker & J. W. ter Vrugt (Philips Lighting Division, Eindhoven): Luminescence of calcium orthophosphates activated with divalent europium.
Philips Res. Repts. **23**, 362-366, 1968 (No. 4).

G. Zanmarchi & C. Haas: Magnon drag at optical frequencies and the infrared spectrum of MnTe.
J. appl. Phys. **39**, 596-597, 1968 (No. 2, Part I). *E*

Inhoud van Electronic Applications **28**, No. 2, 1968:

P. A. Neeteson: Considerations on Schmitt trigger level detectors (p. 41-50).
A. G. W. Uitjens: Recent developments in circuits and transistors for television receivers, VII. Separate e.h.t. supply for colour receivers (p. 51-59).
J. Tuil: Transistor equipped aerial amplifiers (p. 60-78).
G. van Dijk & G. Wolf: Recent developments in circuits and transistors for television receivers, VIII. Simplified high-quality v.h.f. input stages (p. 79-83).

Inhoud van Mullard Technical Communications **10**, No. 92, 1968:

J. Lavallee: Voltage regulation of lower-powered alternators (p. 34-39).
J. Merrett: Voltage transient and dV/dt suppression in thyristor bridges (p. 40-51).
B. J. M. Overgoor: Junction field effect transistors: their structure and operation (p. 52-56).
D. J. G. Janssen: Circuit logic with silicon controlled switches (p. 57-64).
D. Humphries & J. Lavallee: Thyristor control of fan cooling and heating systems (p. 65-72).

Inhoud van Valvo Berichte **14**, No. 3, 1968:

D. Eckstein, C. Lembke & E. Stäbler: Die Entwicklung von Kapazitätsdioden für die elektronische Abstimmung im VHF- und UHF-Bereich (p. 83-98).
J. Koch: Berechnung der Kapazität von Spulen, insbesondere in Schalenkernen (p. 99-119).

Jaargang 29, 1968, no. 11 blz. 333-360 Verschenen 3 december 1968

Digitized in Heiloo the Netherlands Not for sale!